Departamento de Ingeniería Telemática

PROYECTO FIN DE CARRERA

Aplicación para gestionar un equipo de fútbol en plataforma Android

Autor: José Antonio Jiménez Moreno

Tutor: Vicente Luque Centeno

Leganés, Enero de 2016

Agradecimientos

Una vez realizado este proyecto y con ello mis estudios universitarios después de tanto tiempo me gustaría agradecer:

A mis padres, por su sacrificio y esfuerzo para que yo pudiera realizar estudios universitarios y tener más oportunidades en el mundo laboral. Sé que desde ahí arriba mi padre estará orgulloso.

A mi hermana, por tener que aguantarme muchas horas mientras le enseñaba mis avances en la carrera y en el proyecto.

A mi familia por apoyarme y haber sido pacientes todo este tiempo.

A mis compañeros y amigos, sin ellos la vida en la universidad hubiera sido totalmente diferente.

A mi tutor Vicente, por haberme guiado en la realización del proyecto y la memoria del mismo.

Y en especial a Yolanda, por aguantarme en estos últimos meses, saber sacarme una sonrisa en los momentos malos y por tirar de mí hacia delante, sin su apoyo hubiera sido mucho más difícil terminar este proyecto.

Resumen

Hoy en día hay miles de aplicaciones móviles, de todo tipo como juegos, aplicaciones para edición de fotos, consulta de noticias, consulta del tiempo meteorológico, etc… Este hecho era impensable unos años atrás, lo cual demuestra que cada vez más, el uso de los Smartphone está desbancando al uso de ordenadores. Esto se debe a la facilidad de sacar el Smartphone del bolsillo y poder consultar cualquier cosa o entretenerse durante horas.

Cada usuario tiene sus propias necesidades lo que hace que cada vez haya más variedad de aplicaciones móviles adaptadas a todo tipo de personas sin importar la edad.

De ahí nace la idea de este proyecto. Realizar una aplicación que me permitiera introducirme en el desarrollo de aplicaciones y en el sistema operativo Android, ya que es el sistema operativo con mayor crecimiento de los últimos años.

El desarrollo de la aplicación se hace en lenguaje de programación Java, propio de Android y con el entorno de desarrollo propio de Google Android Studio.

La aplicación está orientada a todo aquel que quiera realizar una gestión de su equipo de fútbol o baloncesto en la palma de su mano, se pueden registrar jugadores, equipos, jornadas; editarlos y llevar un seguimiento de todas las jornadas del calendario con su resultado fechas, lugar del encuentro, etc…

Se ha intentado seguir los patrones de diseño recomendados por Google diseñando una interfaz de usuario sencilla y cómoda con una gran armonía entre pantallas. Debido a la gran afición por el diseño gráfico se han realizado todos y cada uno de los diseños de iconos, pantallas, figuras…

Otro punto extra que se le ha dado a la aplicación y así estar a la última en el desarrollo de aplicaciones es que se ha añadido la compatibilidad con smartwatch.

Palabras clave: Android, Smartphone, Smartwatch, wear, wearable, móvil, móviles, aplicación, aplicaciones, app, apps, equipo, fútbol, baloncesto.

Abstract

Nowadays, there are thousands of mobile applications, such as games, applications to edit pictures, newsletters apps, weather apps… This situation was unbelievable years ago, and it means that, little by little, Smartphones are being used more than PCs. This is due to the ease of getting the Smartphones of our pocket to check anything or to be entertained for hours.

Every user has his own needs, and that's why there are more and more mobile apps in the market, adapted to all people without taking into account their age.

Hence the idea for this Project: To create an application that allow me to go deep into the development of mobile applications and into Android operating system. Android is the fastest growing operating system in recent years.

The development of this application is made in Java language, typical of Android, and with the development environment Google Android Studio.

The app is faced to people who wants to manage easily their own football or basketball team. Users can register players, teams, journeys… They can edit the information, in order to manage and to follow the current situation of their team, by updating the results of the matches, the dates of the matches (in case they change), the places where matches are played…

It has been tried to follow the design patterns recommended by Google to get a simple and a comfortable interface for the users, with a great screens harmony. Because of the great love for graphic design, each icon designs, displays, figures… have been made specifically for this Project.

Last but not least, we have to mention that this application is fully compatible with Smartwatch, thus be the latest in mobile applications development.

Keywords: Android, Smartphone, Smartwatch, wear, wearable, mobile, application, applications, app, apps, team, football, basketball.

Contenido

Índice de figuras

Índice de tablas

1. Introducción y objetivos

En este primer apartado de la documentación se pretende dar una visión general sobre el proyecto realizado, el planteamiento inicial y los objetivos que se pretenden cumplir con la realización de la aplicación.

1.1 Planteamiento inicial

Las aplicaciones móviles sirven para satisfacer las necesidades de los usuarios, existen aplicaciones que sólo buscan entretener, pero las hay de todo tipo, como por ejemplo retocar fotos, consultar el tiempo, el tráfico, el horario de autobuses, la clasificación de tu equipo de fútbol y un sinfín más de tipologías.

En un principio se decide crear una aplicación de uso personal, que permitiera la gestión de un equipo de fútbol sala propio. De esta forma, se podría acceder fácilmente, así como tener en un mismo entorno, toda la información relativa a los jugadores del equipo, los datos de estos jugadores, el calendario completo de la liga, los equipos rivales contra los que se enfrentarían….

Esta aplicación inicial iba a ser cerrada, incluyendo sólo los datos relativos a un equipo, por lo que una vez diseñada, la información no podría ser modificada por el usuario posteriormente. Pero, ¿qué pasaría si se cambiaba algún partido de fecha o de lugar?, ¿y si algún equipo se retiraba y el calendario variaba?

Fue entonces cuando se decide darle una vuelta a la idea inicial y hacer una aplicación para “todos”, es decir, una aplicación que no tuviera datos cerrados relativos sólo a un equipo, sino que estos datos pudieran ser modificados por el usuario. De esta forma, cualquier entrenador o jugador podría tener los datos de su equipo y actualizarlos sin necesidad de acudir al diseñador/programador de la aplicación. Así, se consigue que el usuario tenga total autonomía para actualizar los resultados de los partidos, añadir o borrar jornadas, modificar su fecha, etc.

Una aplicación móvil trata de eso, de ser útil y llegar al mayor número de usuarios posible, y por eso se decidió hacer una aplicación flexible y que pudiera ser utilizada por cualquier jugador o entrenador del equipo que fuera.

Para concluir, y con el fin de estar al día con las nuevas tecnologías, se decide hacer la aplicación compatible con Smartwatch. De esa forma, el usuario podrá tener toda la información relativa a su próximo partido simplemente con girar la muñeca y consultar la próxima jornada.

1.2 Motivación

A continuación se explicarán los motivos por los cuales se decide realizar el PFC con el desarrollo de una aplicación Android.

Después de trabajar varios años como analista programador Java, sentía curiosidad por nuevas tecnologías. Un día decidí leer artículos sobre programación en Android, y poco a poco iba sintiendo más curiosidad ya que la programación Android cubría también otra de mis pasiones: el diseño gráfico. Tengo gran interés en el diseño por ordenador. Esto me hizo querer profundizar en el diseño de aplicaciones Android, pudiendo aunar la programación “fuera” del trabajo y el diseño de pantallas e iconos que un usuario pudiera ver.

Empecé realizando pequeñas aplicaciones, pero muy centradas en el diseño, iconos bonitos y coloridos pero sin gran utilidad. Fue entonces cuando se me ocurrió crear una aplicación útil, bonita a la vista y por qué no, que también fuera suficientemente completa para llamarlo “proyecto”. Añadiendo otro ingrediente a la receta, mi afición por el fútbol sala, hice un pequeño boceto de la aplicación y la enseñé a los jugadores de mi equipo; todos me pedían que la compartiera con ellos, y fue entonces cuando me dí cuenta que lo que estaba haciendo iba más allá de un entretenimiento, estaba haciendo algo que además, parecía gustar a la gente.

Una vez finalizada la aplicación que atañe este PFC, la volví a mostrar, y la reacción de mis compañeros fue diferente; ahora me decían que podría venderla y sacarle partido a lo que estaba haciendo. ¿Por qué no? ¿Por qué no tomar este proyecto como punto de partida para poder realizar aplicaciones que los usuarios necesiten en su día a día? Es verdad que hay millones de aplicaciones, pero también es cierto que necesidades también.

Busqué entonces alguna aplicación parecida pero no encontré ninguna. Puede que no buscara muy concienzudamente, pero pensé entonces que acababa de hacer algo que no se había pensado antes. Por este motivo, no descarto compaginar mi trabajo actual y dedicarme al desarrollo de aplicaciones móvil para dispositivos Android.

Cómo apasionado a la tecnología, me gusta estar a la última. Todos los “cacharros” nuevos me encantan, y de ahí que integrara un dispositivo wearable en la aplicación y así seguir innovando en nuevas tecnologías.

1.3 Objetivos

El objetivo principal de este proyecto es la realización de una aplicación para dispositivos móviles que permita:

- Gestionar un equipo de fútbol o baloncesto, pudiendo crear, editar y borrar jornadas, jugadores, equipos, etc...
- Conocer en cualquier momento y sin necesidad de estar conectado a internet el calendario completo de la liga, resultados de jornadas anteriores, información de los jugadores de tu equipo como la edad, el dorsal, goles o puntos anotados, tarjetas amarillas, etc…
- Consultar la información de la delegación correspondiente a tu competición dentro de la propia aplicación.
- Información de equipos rivales como nombre, color de la camiseta o incluso número de teléfono para contactar con su delegado.
- Se podrá consultar toda esta información como jornadas, resultados, jugadores y equipos rivales en un Smartwatch sincronizándose con el Smartphone.

La aplicación deberá crearse usando lenguaje Java, para funcionar sobre el sistema operativo Android, la aplicación también será compatible con dispositivos wearables para una rápida y cómoda visualización de los datos correspondientes al equipo, de una manera rápida y sencilla sin necesidad de utilizar el Smartphone.

1.4 Estructura del documento

En este apartado se describe la estructura del documento para una fácil lectura e interpretación.

El documento se estructura en seis apartados importantes que son introducción, el estado del arte, análisis, diseño de la aplicación y conclusiones.

En el primer apartado **Introducción y objetivos,** se muestra el planteamiento e idea principal de la aplicación a realizar, la motivación por la cual se decide hacer una aplicación como proyecto final de carrera y los objetivos que se desean conseguir con la realización de la aplicación.

En el segundo apartado de este documento se realiza un análisis del sector referente a la realización de este proyecto, en él se hará una introducción a Android, tecnología utilizada para la realización de la aplicación, así como su

diseño y la estructura de las aplicaciones Android. También se describe el entorno Android Studio utilizado para el desarrollo de la aplicación.

En el tercer apartado se realiza un análisis de los requisitos que debe cubrir la aplicación, identificando los casos de uso necesarios para cubrirlos y más adelante se detallan todos y cada uno de los casos de uso ya sea de la aplicación móvil como de la aplicación wear.

En el cuarto y más amplio apartado se detalla el diseño utilizado para la creación de la aplicación, en primer lugar se hace una descripción detallada de las tres partes principales, que son la definición del modelo de datos, la aplicación móvil y la aplicación wear. Una vez detallada la aplicación se muestran todos los detalles, mediante **diagramas de clase**, explicando la estructura utilizada, **diagramas de flujo** indicando el flujo que sigue la aplicación interactuando con el usuario y los **diagramas de secuencia** donde se entra más en detalle técnico de los métodos y clases que intervienen en esos flujos.

En el quinto apartado se detalla el diseño gráfico utilizado para la interfaz de usuario, explicando y mostrando los iconos, listados y pantallas diseñadas.

En el sexto y último apartado se hace un balance de la aplicación donde se muestran las conclusiones obtenidas con la realización de este proyecto y las posibles mejoras de cara a mejorarla en un futuro.

2. Estado del arte

2.1 Android

Android es un sistema operativo móvil, diseñado para controlar dispositivos móviles como Smartphones, Tablets o PDAs; En la actualidad también se está integrando Android en televisores y coches.

La primera versión de Android fue desarrollada por la compañía Android Inc., fundada en 2003. La evolución del trabajo de esta compañía se llevaba a cabo en secreto hasta que el gigante Google empezó a mostrar interés en el negocio de los dispositivos y comunicaciones móviles y en 2005 adquirió la compañía. Una vez en manos de Google, el equipo desarrolló una plataforma para dispositivos móviles basada en el núcleo Linux promocionándose a fabricantes de dispositivos como un sistema flexible y actualizable.

En noviembre de 2007 finalmente se anunció el sistema operativo al mundo por parte de la Open Handset Alliance, un consorcio creado con más de 70 empresas del mundo de las telecomunicaciones como HTC, Qualcomm, Intel, Motorola, LG, etc…). Su principal objetivo era el desarrollo de estándares abiertos para dispositivos móviles. Bajo esta premisa, Google liberó la mayor parte del código fuente del sistema operativo Android usando la licencia Apache, una licencia libre, gratuita y de código abierto.

La evolución del sistema operativo hasta la actualidad ha sido espectacular llegando a una cuota de mercado en el año 2014 del 81,5% según datos de la International Data Corporation (IDC). Posteriores versiones a su lanzamiento han ido añadiendo funcionalidades a los smartphones como la pantalla multi-táctil, reconocimiento facial, compatibilidad con otras tecnologías punteras como HTML5 y Adobe Flash, soporte de NFC (Near Field Communication), grabación de vídeo 3D...

2.2 Diseño en Android

El sistema operativo Android está compuesto de un núcleo Linux, sobre el que se ejecutan: por una parte, la máquina virtual Dalvik, basada en Java pero diseñada más eficientemente para dispositivos móviles, y por otra parte una serie de librerías o bibliotecas programadas en C y C++ que ofrecen acceso principalmente a las capacidades gráficas del dispositivo. Una capa por encima encontramos el marco de trabajo de aplicaciones, una colección de librerías Java

básico, adaptadas para su ejecución sobre Dalvik; finalmente tenemos las aplicaciones Java, que son las que ofrecen la funcionalidad al usuario.

Figura 1: Capas de aplicación android

A continuación se detallan cada uno de los componentes que conforman el sistema operativo Android:

- **Applications (aplicaciones):** son aplicaciones base que incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas, contactos, navegador y otros. Todas escritas en lenguaje Java.
- **Application framework (marco de trabajo):** Los desarrolladores tienen acceso completo a los APIs del framework de las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra hacer uso de estas.
- **Libraries (bibliotecas):** Android incluye un conjunto de bibliotecas de C/C++ usadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del framework de aplicaciones de Android. Las bibliotecas escritas en lenguaje C/C++ incluyen un administrador de pantalla táctil (surface manager), un framework OpenCore (para el aprovechamiento de las capacidades multimedia), una base de datos relacional SQLite, una API gráfica OpenGL ES 2.0 3D, un

motor de renderizado WebKit, un motor gráfico SGL, el protocolo de comunicación segura SSL y una biblioteca estándar de C, llamada "Bionic" y desarrollada por Google específicamente para Android a partir de la biblioteca estándar "libc" de BSD.

- **Android runtime (funcionalidad en tiempo de ejecución):** Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente. Dalvik ejecuta archivos en el formato Dalvik Executable (.dex), el cual está optimizado para memoria mínima. La Máquina Virtual está basada en registros y corre clases compiladas por el compilador de Java que han sido transformadas al formato dex por la herramienta incluida "dx".
- **Linux Kernel (Núcleo Linux):** Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software [1].

2.3 Estructura básica aplicación Android

Existen una serie de componentes que son los más utilizados en el diseño y la creación de una aplicación Android, a continuación se detallarán los más importantes:

- **Vista (View):** Las vistas son los elementos que componen la interfaz de usuario de la aplicación como por ejemplo, botones, listados, entradas de texto. Todas las vistas van a ser objetos que heredan de la clase View, por lo que pueden ser definidas utilizando código Java. Lo habitual será definir las vistas utilizando ficheros XML y así dejar al sistema que cree los objetos por nosotros a partir de este fichero.
- **Layout:** Los layouts son conjuntos de vistas agrupadas de una determinada manera. Existen diferentes tipos de layouts como son ***Relativelayout***, ***Linearlayout***, etc... que permiten organizar las vistas de forma lineal, en cuadrícula con posición absoluta, etc…
- **Actividad (Activity):** Las aplicaciones Android estarán formadas por un conjunto de elementos básicos de visualización. Cada uno de estos elementos, o pantallas se conoce como actividad. Su función principal es la creación de la interfaz de usuario. Una aplicación suele estar compuesta por varias actividades. Las diferentes actividades creadas

serán independientes entre sí, aunque todas tendrán un objetivo común. Todas las actividades extienden de Activity.

- **Servicio (Service):** Los servicios son procesos que se ejecutan en background, sin necesidad de interacción con el usuario. Es parecido a un demonio en Unix o un servicio en Windows. Android dispone de dos tipos de servicios: locales y remotos. Los servicios locales son ejecutados en el mismo proceso, y los remotos en procesos separados. Corresponde a cada servicio implementar el comportamiento adecuado, habitualmente creará un hilo de ejecución (thread) secundario donde se realizará el trabajo.
- **Intención (Intent):** Una intención, como su nombre indica, representa la voluntad de realizar alguna acción, como realizar una llamada de teléfono, visualizar una página web, se utiliza cada vez que queramos lanzar una actividad, un servicio, lanzar un anuncio de broadcast o comunicarnos con un servicio. Uno de los usos más frecuentes en las intenciones es para la comunicación de información entre los componentes de la aplicación.
- **Fragment:** Con la llegada de aparatos con pantallas grandes, ya sean tablets o incluso televisiones, se introduce un nuevo elemento como es el Fragment, se utiliza para controlar un bloque funcional en la interfaz de usuario y así poder combinar varias funcionalidades dentro de la misma pantalla.
- **Receptor de anuncios (Broadcast receiver):** Un receptor de anuncios recibe y reacciona ante anuncios de tipo broadcast, estos anuncios pueden ser originados por el sistema o por las aplicaciones como por ejemplo, batería baja, llamada entrante, etc ... Las aplicaciones también pueden crear y lanzar nuevos tipos de anuncios broadcast. Estos receptores no tienen interfaz gráfica aunque pueden lanzar una actividad si se desea.
- **Proveedores de Contenido (Content Provider):** Android define un mecanismo estándar para compartir datos entre aplicaciones sin necesidad de comprometer la seguridad del sistema de ficheros. Con este mecanismo podemos acceder a información de otras aplicaciones como la lista de contactos.

Una vez detallada la estructura de una aplicación Android a nivel de elementos que la componen, a continuación se detalla la estructura de un proyecto Android, el entorno de trabajo utilizado para este proyecto es Android Studio por lo que se detalla la estructura creada por este IDE, no existen muchas diferencias con otros entornos de trabajo como Eclipse.

El fichero más importante de una aplicación Android es el *AndroidManifest.xml,* este fichero describe la aplicación Android. En él se indica toda la información de la aplicación como las actividades que la componen, las intenciones, los servicios y los proveedores de contenido. También se declaran

los permisos que se le dan a una aplicación, se indica el rango de versiones de Android que podrán ejecutar la aplicación, el paquete Java, la versión de la aplicación, etc…

Las carpetas que se crean en un proyecto Android son las siguientes:

- **manifests**: Carpeta que contiene el fichero *AndroidManifest.xml*
- **java**: Carpeta con el código fuente de la aplicación. Los ficheros java se almacenan en carpetas según el nombre de su paquete.
- **res**: Carpeta que contiene los recursos usados por la aplicación:
  - **drawable**: En esta carpeta se almacenan los ficheros de imágenes (JPG o PNG) y descriptores de imágenes en XML.
  - **layout**: Contiene ficheros XML con las vistas que componen la aplicación.
  - **menú**: Ficheros XML con los menús de cada actividad.
  - **values**: Incluye ficheros XML que se utilizarán para indicar los valores usados en la aplicación, algunos ejemplos de estos ficheros serán *strings.xml* donde se incluirán todas las cadenas de texto utilizadas por la aplicación, otro fichero muy común es *dimens.xml* donde se definirán las medidas de los elementos visuales de la aplicación.
  - **anim**: Se incluirán las animaciones que utilizarán los distintos elementos de la aplicación.

Un proyecto Android también permite crear subcarpetas para diferenciar tamaños y densidad de píxeles en las diferentes pantallas “drawable-ldpi” ó “drawable-hdpi”. Otras subcarpetas importantes que se pueden necesitar son las relacionadas con los diferentes idiomas, Android incorpora sufijos a las carpetas como “res/values-en” y “res/values-de” para guardar traducciones en inglés y alemán de los textos de la aplicación [2].

2.4 Patrón de diseño MVC

Para el diseño y la implementación de aplicaciones Android el patrón más utilizado es el MVC (Modelo, Vista Controlador) el cual se detalla a continuación [3]:

Modelo Vista Controlador, o MVC, es un patrón de arquitectura de software que separa una aplicación en tres capas o componentes diferenciados: El Modelo representa los datos de la aplicación, la Vista es el equivalente a la interfaz de usuario, y el tercer componente es el Controlador, el cual contiene la lógica de procesamiento de las acciones que realiza el usuario en el ciclo de vida de la aplicación.

Este patrón permite un “desacoplamiento” entre los tres elementos, es decir, la definición y detalles de cada uno de ellos más o menos desconocidos para los otros dos. A continuación se detalla cada componente por separado:

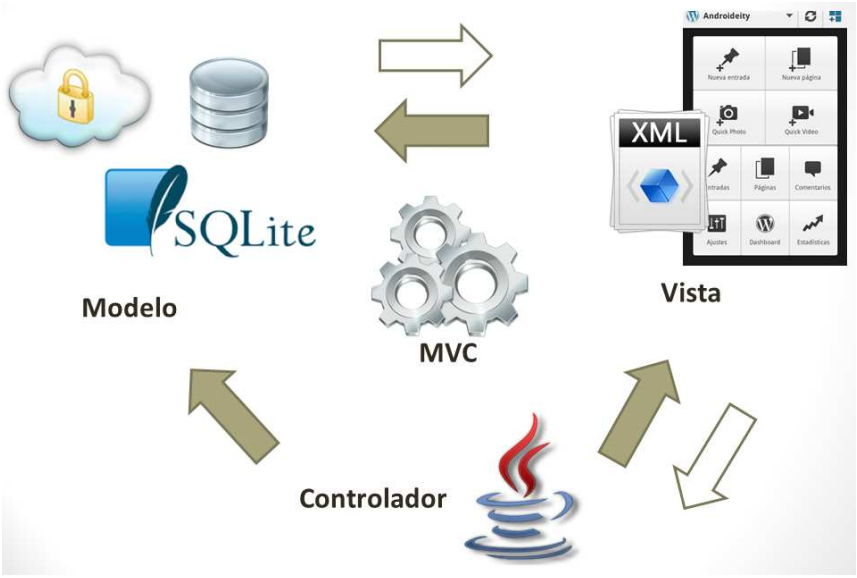

Figura 2: Patrón MVC Android

- **Modelo:** Representa los datos con los que trabaja la aplicación y, en muchas ocasiones, también incluye la lógica de negocio utilizada para manejarlos. Cada entidad del modelo será consultada o actualizada por la Vista y el Controlador respectivamente, pero sin llegar a tener conocimiento de ello. Este componente al ser definido al inicio del diseño, no suele sufrir cambios a lo largo de la implementación de la aplicación.
- **Vista:** Este componente se encarga de recoger la información del Modelo y presentarla al usuario de forma que pueda ser interpretada con facilidad y manejada por el mismo, casi siempre a través de una interfaz gráfica. Engloba toda la lógica de interacción entre el usuario y los métodos de entrada permitido por los dispositivos físicos como pueda ser un teclado, pantallas sensibles al tacto, etc…) incluyendo la generación de eventos necesarios que se enviarán al Controlador. Gracias a este desacoplamiento entre componentes un mismo Modelo puede ser mostrado al usuario de diferentes maneras e interfaces.
- **Controlador:** Recibe los eventos generados por el usuario en la Vista, los procesa y accede al Modelo para generar las respuestas adecuadas. Puede producirse a veces que el flujo entre usuario y aplicación no implique el uso de los tres componentes, se puede dar el caso que la acción del usuario implique únicamente a la Vista y el Controlador, en este caso el Controlador es el encargado de procesar la acción y modificar la Vista para que refleje estos cambios.

2.5 Aproximación al patrón MVC

El patrón MVC “puro”, Vista y Controlador deben mantener un alto grado de desacoplamiento. El Controlador no interviene en la actualización de la Vista si fuera necesario, dejando que ella misma detecte los cambios en el Modelo, mediante eventos o consulta directa. La Vista tampoco incluye la lógica para gestionar los eventos del usuario, estos se reciben directamente por el Controlador. Mantener este grado de desacoplamiento no es fácil. Debido a esto, basándose en MVC se han ideado múltiples patrones derivados que trasladan hacia un lado (Vista) u otro (Controlador) el peso de la lógica principal del a aplicación, según convenga al escenario.

Existen situaciones en los que la Vista se encarga de las acciones del usuario y no el Controlador; mientras que el Controlador se encarga del tratamiento de los datos y de adaptarlos en formato para que la Vista pueda procesarlos de manera fácil, el Controlador no conocerá directamente las acciones del usuario, sólo las que la Vista delegue sobre él [3] [4].

En otras variantes la Vista no tiene acceso directo al Modelo. La solución es que el modelo recoja los datos del Modelo y se los proporcione a la Vista, quedando esta como una capa gráfica controlada por el Controlador.

Estas variantes son las que mejor se adaptan al patrón utilizado por aplicaciones Android, debido a la dificultad de desacoplar los componentes, de hecho la Vista y el Controlador tienen un alto acoplamiento (Activity-View), este acoplamiento se verá más adelante mediante la explicación del diseño de la aplicación que nos ocupa.

2.6 Entorno de desarrollo

Para la realización de la aplicación se utilizará el entorno de desarrollo proporcionado por Google AndroidStudio:

Figura 3: Android Studio

Es un entorno de desarrollo integrado para la plataforma Android. Se anunció el 16 de Mayo de 2013 en la conferencia Google I/O. Este entorno reemplazó a Eclipse como IDE oficial para el desarrollo de aplicaciones Android. La primera versión estable fue publicada en Diciembre de 2014.

Está basado en el software IntelliJ IDEA de JetBrains. Está disponible para las plataformas Microsoft Windows, Mac OS X y GNU/Linux.

Entre sus características destacan la renderización en tiempo real, consola de desarrollador donde se muestran consejos de optimización, ayuda a la traducción, etc… Da soporte para construcción basada en Gradle dónde importar librerías de forma automática. Ofrece plantillas para crear diseños comunes de Android y otros componentes. Una de las características por las cuales se decidió usarlo fue el soporte para programar aplicaciones para Android Wear.

Su interfaz es muy intuitiva y permite crear una aplicación Android en pocos y sencillos pasos, simplemente seleccionando el nombre de la aplicación (Figura 4), las versiones de Android compatibles y el tipo de aplicación (Figura 5) y el formato de la interfaz de usuario (Figura 6):

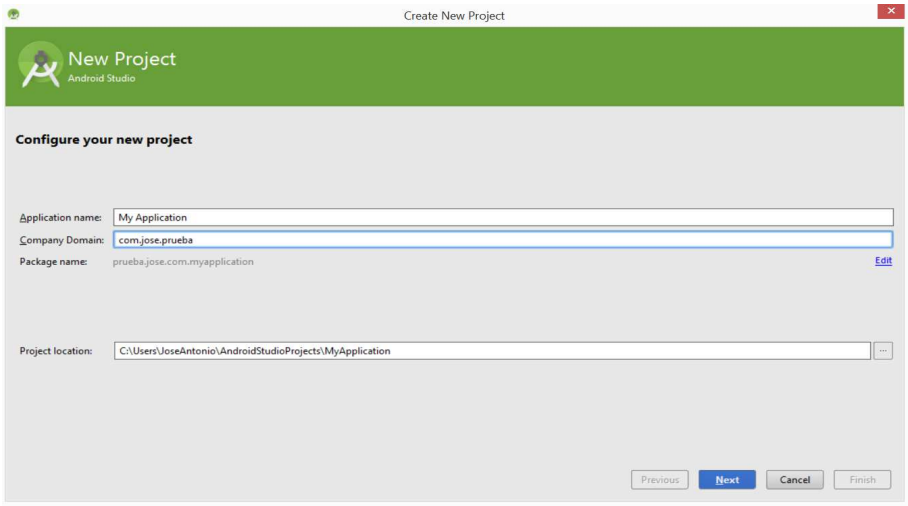

Figura 4: Paso 1 creación aplicación, selección nombre

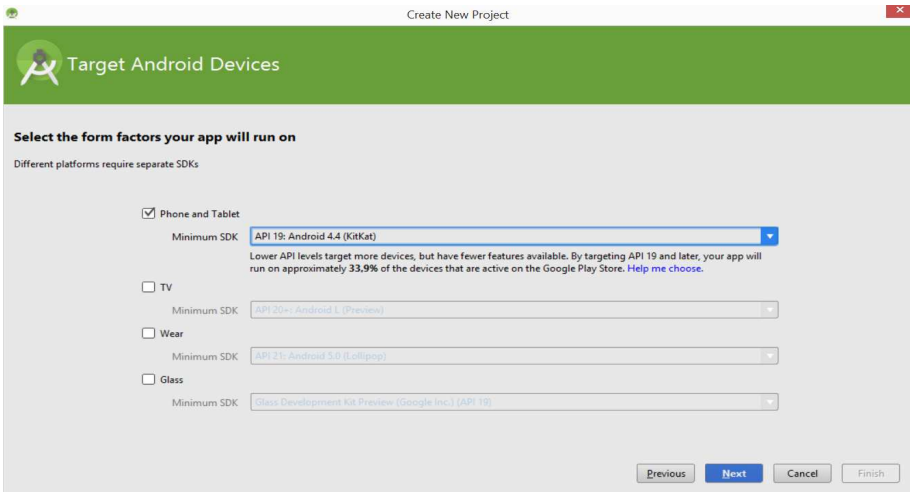

Figura 5: Paso 2 creación aplicación, selección versiones compatibles y tipo de aplicación

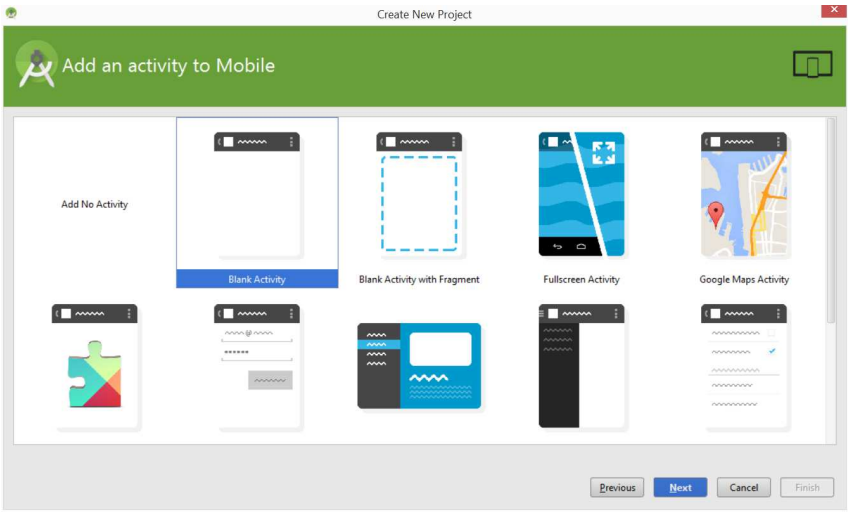

Figura 6: Formato interfaz de la aplicación

Este entorno integrado proporciona multitud de ayudas, cabe destacar entre ellas la visualización en tiempo real del diseño de la interfaz gráfica en distintos dispositivos y densidades de pantalla:

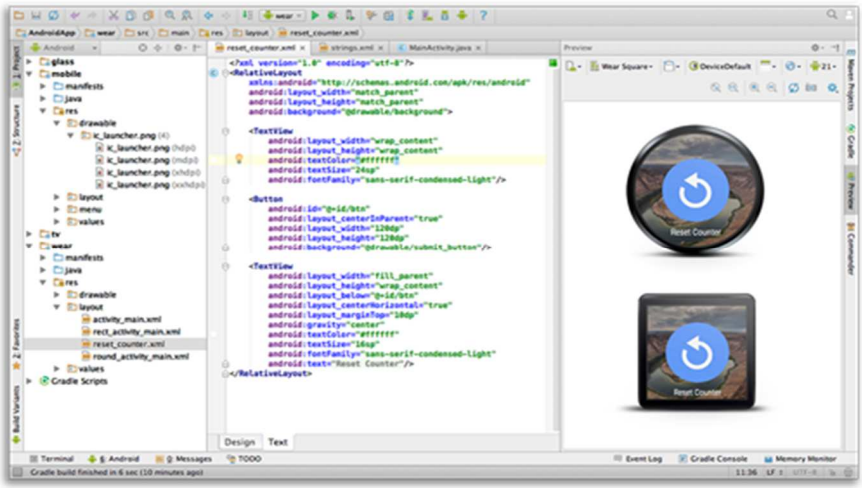

Figura 7: Ayudas en el diseño de la interfaz

Otra gran utilidad del entorno es la posible depuración de código mediante un emulador integrado (AVD Manager), que permite ejecutar la aplicación en un dispositivo virtual, configurable por el usuario, definiendo el tamaño y densidad de pantalla, versión de Android, tipo de dispositivo (Smartphone, Smartwatch, TV, Tablet…), este emulador permite realizar cualquier prueba que se desee como si de un dispositivo físico se tratase:

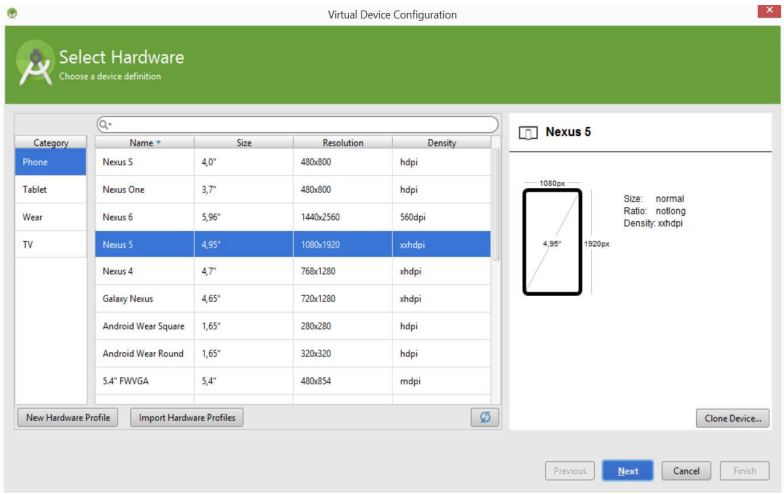

Figura 8: Configuración emulador Android

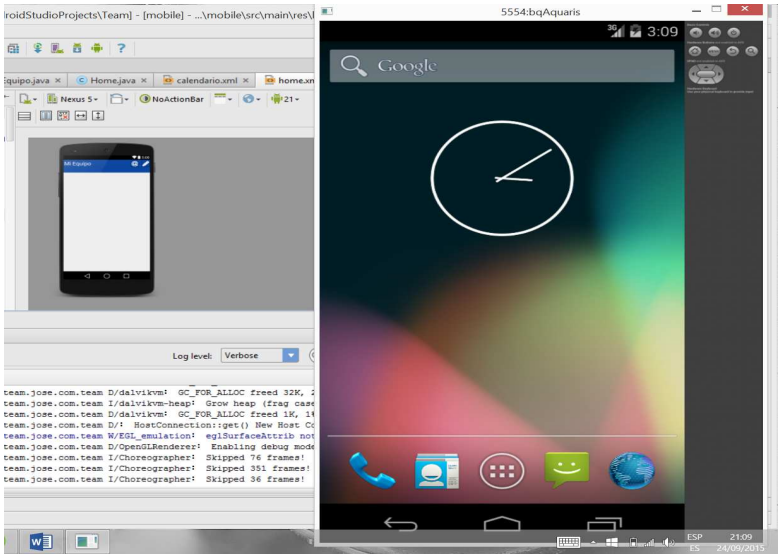

Figura 9: Visualización emulador Android

Existen otros entornos de trabajo como el citado anteriormente (Eclipse), se elija el entorno que se elija para desarrollar aplicaciones es necesario el kit de desarrollo de software Android SDK (Software Development Kit) el cual incluye el conjunto de herramientas básicas que permiten compilar y depurar aplicaciones escritas para el sistema operativo Android, así como empaquetar y firmar las aplicaciones para su posterior distribución (por ejemplo Play Store), este kit viene integrado ya en el IDE Android Studio (SDK Manager) el cual permitirá descargar, instalar y actualizar las revisiones necesarias para desarrollar aplicaciones para las diferentes versiones Android [5].

3. Análisis

3.1 Requisitos

En este apartado se analizarán las necesidades y se identificarán los requisitos que deberá satisfacer la aplicación. Los requisitos se dividirán en dos “categorías”, requisitos de la aplicación del Smartphone y la aplicación wearable.

3.1.1 Aplicación Smartphone

A continuación se detallan los requisitos de la aplicación Smartphone, se divide en dos apartados, los requisitos de la interfaz gráfica y los requisitos de la aplicación.

3.1.1.1 Interfaz Gráfica

Los requisitos de la interfaz gráfica para Smartphone son los siguientes:

RG.AS001 – La aplicación móvil deberá mostrar de forma intuitiva todas las entidades de la aplicación (Jornadas, Jugadores, Equipos).

RG.AS002 – Se podrá añadir una entidad desde la pantalla principal de una manera intuitiva, se añadirá a cada listado de las entidades un botón de añadir.

RG.AS003 – Desde la pantalla principal se podrá acceder a la edición de cualquier entidad, del propio equipo y a la página web.

RG.AS004 – La visualización de las entidades se hará de forma simple e intuitiva.

RG.AS005 – Ante cualquier error la aplicación deberá notificarse al usuario de forma clara.

RG.AS006 – Desde la pantalla de visualización de cada entidad se podrá editar directamente cualquier dato de la propia entidad.

RG.AS007 - Desde la pantalla principal se podrá borrar cualquier entidad.

3.1.1.2 Aplicación

Los requisitos de la aplicación para Smartphone son:

RA.AS001 – Se podrán añadir entidades en cualquier momento.

RA.AS002 – Se podrán editar entidades y borrarlas en cualquier momento.

RA.AS003 – Al editar cualquier entidad se deberá comprobar que la edición es correcta y no producirá ningún error, si la comprobación fuera negativa se deberá notificar al usuario de forma clara e intuitiva el error producido.

RA.AS004 – Al mostrar los listados de las entidades se deberán ordenar según los siguientes criterios:

- Jornadas: Por número de jornada
- Jugadores: Por posición
- Equipos: Por orden de creación

RA.AS005 – La página web a mostrar se podrá modificar tantas veces como se quiera. Al actualizar la url se deberá mostrar la nueva página web.

3.1.2 Aplicación wearable

A continuación se detallan los requisitos de la aplicación wearable, igualmente se dividen en requisitos de interfaz gráfica y requisitos de aplicación.

3.1.2.1 Interfaz Gráfica

Los requisitos de la interfaz gráfica para la aplicación wearable son los siguientes:

RG.AW001 – La visualización de jornadas, jugadores y equipos se hará en forma de listado acorde al diseño recomendado por Google [6].

RG.AW002 – La visualización de cada entidad se hará de forma simplificada e intuitiva.

3.1.2.2 Aplicación:

Los requisitos de la aplicación wearable son los siguientes:

RA.AW001 – Si no hay entidades que mostrar la aplicación lo notificará al usuario.

RA.AW002 – Los listados se mostrarán de forma ordenada al igual que en la aplicación móvil.

RA.AW003 – La aplicación wearable podrá solicitar información a la aplicación móvil sin necesidad que se esté ejecutando en el Smartphone.

3.2 Casos de uso

Una vez analizadas las necesidades e identificados los requisitos de la aplicación se procederá a identificar y definir los casos de uso necesarios para cubrir estos requisitos.

3.2.1 Aplicación Smartphone

A continuación se listan los casos de uso que interactúan directamente con el usuario en la aplicación del Smartphone, los casos de usos se agruparán en 3 bloques, que son la creación, visualización y edición de las entidades de la aplicación:

- CU-AS001: Creación equipo propio.
- CU-AS002: Creación jugador.
- CU-AS003: Creación equipo rival.
- CU-AS004: Creación jornada.

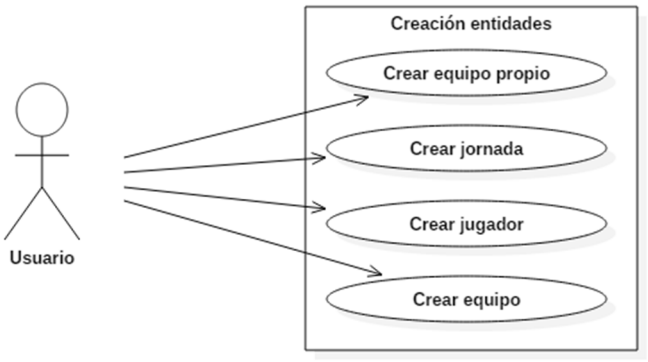

Figura 10: Casos de uso creación entidades

- CU-AS005: Visualización calendario.
- CU-AS006: Visualización jugadores.
- CU-AS007: Visualización equipos.
- CU-AS008: Visualización goleadores/anotadores.

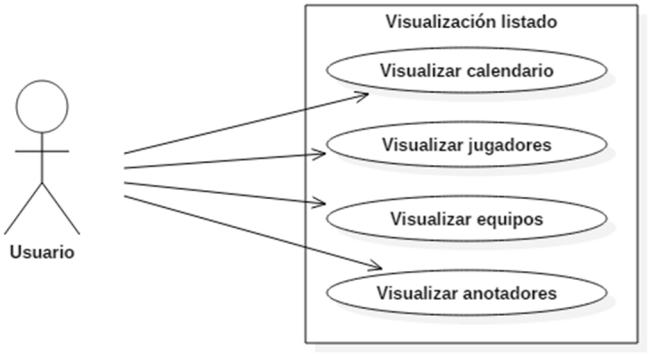

Figura 11: Casos de uso visualización listados entidades

- CU-AS009: Visualización equipo propio.
- CU-AS010: Visualización jugador.
- CU-AS011: Visualización equipo rival.
- CU-AS012: Visualización jornada.
- CU-AS013: Visualización página web.
- CU-AS014: Edición equipo propio.
- CU-AS015: Edición jugador.
- CU-AS016: Edición equipo rival.
- CU-AS017: Edición jornada.
- CU-AS018: Edición página web.

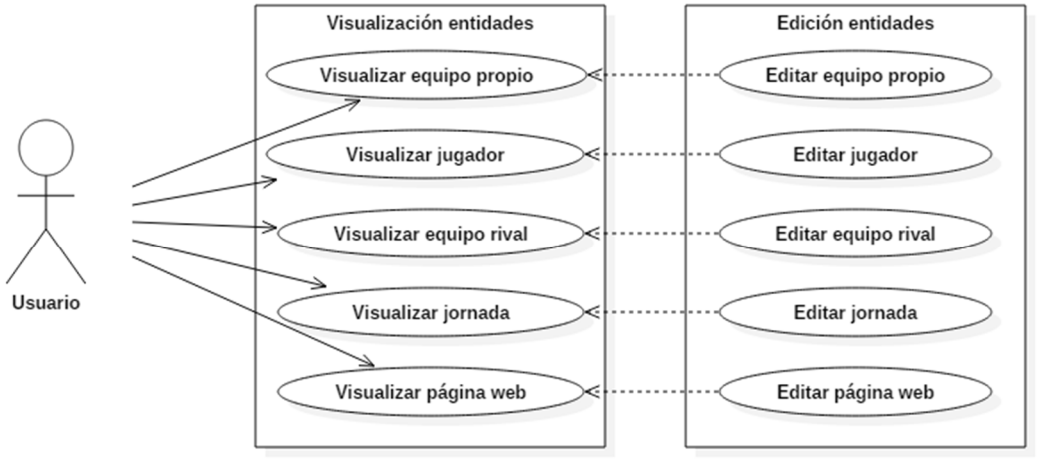

Figura 12: Casos de uso visualización y edición entidades

- CU-AS019: Borrar jornada.
- CU-AS020: Borrar jugador.
- CU-AS021: Borrar equipo.
- CU-AS022: Llamada delegado equipo

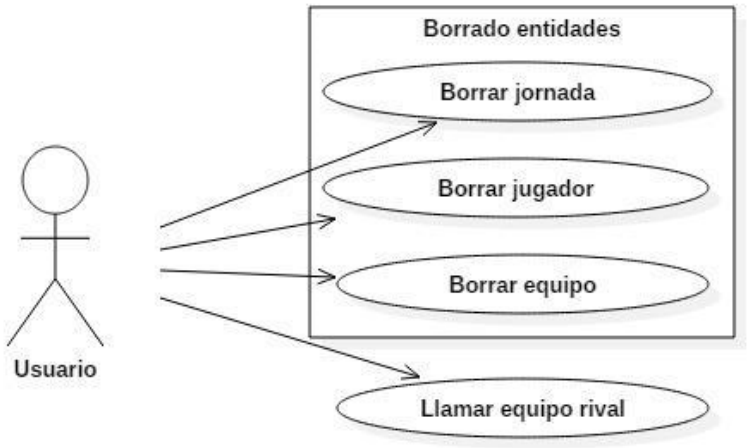

Figura 13: Casos de uso borrado entidades y llamada telefónica

Una vez identificados los casos de uso para la aplicación del Smartphone se procede a detallar cada uno de ellos:

Para una fácil comprensión de cada caso de uso se utilizará una tabla con los siguientes campos:

- Código: Código único que identifica el UC.
- Caso de uso: Nombre del caso de uso.
- Actor: Actor que actúa en el UC.

- Descripción: Breve descripción del UC.
- Precondiciones: Escenario necesario que se debe dar para que se pueda realizar el UC.
- Curso normal: Flujo que se debe seguir para el correcto funcionamiento.
- Excepciones: Posibles errores que se pueden dar en el UC.
- Postcondiciones: Escenario que se dará una vez concluido el flujo normal el UC.

Código	CU-AS001
Caso de uso	Creación equipo propio
Actor	Usuario
Descripción	Al lanzar la aplicación por primera vez, el usuario deberá crear su equipo
Precondiciones	La aplicación no se habrá ejecutado con anterioridad o los datos de la aplicación habrán sido borrados por el usuario
Curso normal	1. El usuario ejecuta la aplicación por primera vez. 2. El usuario seleccionará un color de camiseta. 3. El usuario seleccionará un nombre de equipo. 4. El usuario seleccionará un tipo de deporte. 5. El usuario pulsará el botón de guardar equipo
Excepciones	
Postcondiciones	Si el usuario ha seleccionado todo lo anterior correctamente el equipo será guardado en la aplicación

Tabla 1: CU-AS001 Creación equipo propio

Código	CU-AS002
Caso de uso	Creación jugador
Actor	Usuario
Descripción	El usuario podrá crear un jugador de su equipo
Precondiciones	La aplicación habrá sido ejecutada y tendrá un equipo propio creado
Curso normal	1. El usuario ejecuta la aplicación 2. El usuario irá a la pestaña “JUGADORES” 3. El usuario pulsará el botón de añadir jugador 4. El usuario seleccionará los datos del nuevo jugador (nombre, dorsal, edad y posición) 5. El usuario pulsará el botón de guardar jugador
Excepciones	Si el usuario no selecciona los campos obligatorios(nombre y posición) se le notificará
Postcondiciones	Si el usuario ha seleccionado todo lo anterior correctamente el jugador será guardado en la aplicación y la aplicación retornada a la pestaña “JUGADORES”

Tabla 2: CU-AS002 Creación jugador

Código	CU-AS003
Caso de uso	Creación equipo rival
Actor	Usuario
Descripción	El usuario podrá crear un equipo rival
Precondiciones	La aplicación habrá sido ejecutada y tendrá un equipo propio creado
Curso normal	1. El usuario ejecuta la aplicación 2. El usuario irá a la pestaña “EQUIPOS” 3. El usuario pulsará el botón de añadir equipo 4. El usuario seleccionará los datos del nuevo equipo (nombre, color camiseta, entrenador y teléfono) 5. El usuario pulsará el botón de guardar jugador
Excepciones	Si el usuario no selecciona los campos obligatorios(nombre y color camiseta) se le notificará
Postcondiciones	Si el usuario ha seleccionado todo lo anterior correctamente el equipo rival será guardado en la aplicación y la aplicación retornada a la pestaña “EQUIPOS”

Tabla 3: CU-AS003 Creación equipo rival

Código	CU-AS004
Caso de uso	Creación jornada
Actor	Usuario
Descripción	El usuario podrá crear una jornada
Precondiciones	La aplicación habrá sido ejecutada y tendrá un equipo propio y un equipo rival como mínimo
Curso normal	1. El usuario ejecuta la aplicación 2. El usuario irá a la pestaña “CALENDARIO” 3. El usuario pulsará el botón de añadir jornada 4. El usuario seleccionará los datos de la jornada (equipo local, equipo visitante, fecha, hora y lugar) 5. El usuario pulsará el botón de guardar
Curso Alternativo	1. El usuario ejecuta la aplicación 2. El usuario irá a la pestaña “CALENDARIO” 3. Si no existen jornadas se mostrará un botón central para añadir la primera jornada
Excepciones	Si el usuario no selecciona los campos correctamente (equipo local y equipo visitante) se le notificará
Postcondiciones	Si el usuario ha seleccionado todo lo anterior correctamente el equipo rival será guardado en la aplicación y la aplicación retornada a la pestaña “EQUIPOS”

Tabla 4: CU-AS004 Creación jornada

Código	CU-AS005
Caso de uso	Visualización calendario
Actor	Usuario
Descripción	El usuario podrá visualizar las jornadas del calendario. Cada jornada del listado mostrará la información de número de jornada, estado, fecha y resultado
Precondiciones	La aplicación habrá sido ejecutada y tendrá mínimo una jornada guardada
Curso normal	1. El usuario ejecuta la aplicación 2. El usuario irá a la pestaña “CALENDARIO”
Curso alternativo	1. El usuario ejecuta la aplicación 2. El usuario irá a la pestaña “CALENDARIO” 3. Si no existen jornadas se mostrará un botón central para añadir la primera jornada
Excepciones	Si se produce un error al cargar las jornadas se notificará al usuario
Postcondiciones	Se mostrará un listado con las jornadas

Tabla 5: CU-AS005 Visualización calendario

Código	CU-AS006
Caso de uso	Visualización jugadores
Actor	Usuario
Descripción	El usuario podrá visualizar los jugadores de su equipo. Cada jugador de listado mostrará el nombre, posición y dorsal.
Precondiciones	La aplicación habrá sido ejecutada y tendrá mínimo un jugador guardado.
Curso normal	1. El usuario ejecuta la aplicación 2. El usuario irá a la pestaña “JUGADORES”
Curso alternativo	1. El usuario ejecuta la aplicación 2. El usuario irá a la pestaña “JUGADORES” 3. Si no existen jugadores se mostrará un botón central para añadir el primer jugador
Excepciones	Si se produce un error al cargar los jugadores se notificará al usuario
Postcondiciones	Se mostrará un listado con los jugadores

Tabla 6: UC-AS007 Visualización jugadores

Código	CU-AS007
Caso de uso	Visualización equipos
Actor	Usuario
Descripción	El usuario podrá visualizar los equipos rivales. Cada equipo del listado mostrará el nombre y nombre de entrenador.
Precondiciones	La aplicación habrá sido ejecutada y tendrá mínimo un equipo guardado.
Curso normal	1. El usuario ejecuta la aplicación 2. El usuario irá a la pestaña “EQUIPOS”
Curso alternativo	1. El usuario ejecuta la aplicación 2. El usuario irá a la pestaña “EQUIPOS” 3. Si no existen equipos se mostrará un botón central para añadir el primer equipo
Excepciones	Si se produce un error al cargar los equipos se notificará al usuario
Postcondiciones	Se mostrará un listado con los equipos

Tabla 7: UC-AS007 Visualización equipos

Código	CU-AS008
Caso de uso	Visualización goleadores/anotadores
Actor	Usuario
Descripción	El usuario podrá visualizar los goleadores/anotadores de su equipo. Cada jugador del listado mostrará el nombre, posición y número de goles.
Precondiciones	La aplicación habrá sido ejecutada y tendrá mínimo un jugador guardado.
Curso normal	1. El usuario ejecuta la aplicación 2. El usuario irá a la pestaña “GOLEADORES” o “ANOTADORES” dependiendo del deporte seleccionado
Curso alternativo	3. El usuario ejecuta la aplicación 4. El usuario irá a la pestaña “GOLEADORES/ANOTADORES” 5. Si no existen jugadores esta pantalla se mostrará en blanco
Excepciones	Si se produce un error al cargar los jugadores se notificará al usuario
Postcondiciones	Se mostrará un listado con los goleadores/anotadores del equipo

Tabla 8: CU-AS008 Visualización goleadores/anotadores

Código	CU-AS009
Caso de uso	Visualización equipo propio
Actor	Usuario
Descripción	El usuario podrá visualizar su equipo, en esta pantalla aparecerá el color de la camiseta y el nombre del equipo
Precondiciones	La aplicación habrá sido ejecutada y tendrá el equipo propio guardado.
Curso normal	1. El usuario ejecuta la aplicación 2. El usuario pulsará el botón situado en la barra superior arriba a la derecha
Curso alternativo	
Excepciones	Si se produce un error al cargar los datos del equipo se notificará al usuario
Postcondiciones	Se mostrarán los datos del equipo

Tabla 9: CU-AS009 Visualización equipo propio

Código	CU-AS010
Caso de uso	Visualización jugador
Actor	Usuario
Descripción	El usuario podrá visualizar cada jugador de su equipo, en esta pantalla aparecerá la posición, nombre, edad, dorsal, goles, tarjetas amarillas* y tarjetas rojas*
Precondiciones	La aplicación habrá sido ejecutada y tendrá mínimo un jugador guardado
Curso normal	1. El usuario ejecuta la aplicación 2. El usuario irá a la pestaña de “JUGADORES” 3. El usuario seleccionará un jugador de la lista
Curso alternativo	
Excepciones	Si se produce un error al cargar los datos del jugador se notificará al usuario
Postcondiciones	Se mostrarán los datos del jugador seleccionado

Tabla 10: CU-AS010 Visualización jugador

Código	CU-AS011
Caso de uso	Visualización equipo rival
Actor	Usuario
Descripción	El usuario podrá visualizar cada equipo rival, en esta pantalla aparecerá el color de la camiseta, nombre del equipo, nombre del delegado/entrenador y número de contacto, también aparecerá un botón para realizar llamada
Precondiciones	La aplicación habrá sido ejecutada y tendrá mínimo un equipo guardado
Curso normal	1. El usuario ejecuta la aplicación 2. El usuario irá a la pestaña de “EQUIPOS” 3. El usuario seleccionará un equipo de la lista
Curso alternativo	
Excepciones	Si se produce un error al cargar los datos del equipo rival se notificará al usuario
Postcondiciones	Se mostrarán los datos del equipo rival seleccionado

Tabla 11: CU-AS011 Visualización equipo rival

Código	CU-AS012
Caso de uso	Visualización jornada
Actor	Usuario
Descripción	El usuario podrá visualizar cada jornada, en esta pantalla aparecerán los equipos local y visitante, fecha, hora, lugar y resultado de la jornada
Precondiciones	La aplicación habrá sido ejecutada y tendrá mínimo una jornada guardada
Curso normal	1. El usuario ejecuta la aplicación 2. El usuario irá a la pestaña de “CALENDARIO” 3. El usuario seleccionará una jornada de la lista
Curso alternativo	
Excepciones	Si se produce un error al cargar los datos de la jornada se notificará al usuario
Postcondiciones	Se mostrarán los datos de la jornada seleccionada

Tabla 12: CU-AS012 Visualización jornada

Código	CU-AS013
Caso de uso	Visualización página web
Actor	Usuario
Descripción	El usuario podrá visualizar una página web introducida por él.
Precondiciones	La aplicación habrá sido ejecutada y tendrá un equipo propio guardado
Curso normal	1. El usuario ejecuta la aplicación 2. El usuario pulsará el segundo icono de la barra superior arriba a la derecha
Curso alternativo	1. El usuario ejecuta la aplicación 2. El usuario pulsará el segundo icono de la barra superior arriba a la derecha 3. Si no hay ninguna url cargada en la aplicación se mostrará un botón central para añadir una nueva url
Excepciones	Si se produce un error al cargar la página web se notificará al usuario
Postcondiciones	Se mostrarán la página web introducida por el usuario

Tabla 13: CU-AS013 Visualización página web

Código	CU-AS014
Caso de uso	Edición equipo propio
Actor	Usuario
Descripción	El usuario podrá editar su equipo
Precondiciones	La aplicación habrá sido ejecutada y creado el equipo propio
Curso normal	1. El usuario ejecuta la aplicación 2. El usuario pulsará el botón de edición de equipo situado en la barra superior 3. El usuario modificará los datos del equipo (color camiseta y nombre) 4. El usuario pulsará el botón de guardar
Curso alternativo	
Excepciones	Si el usuario edita los datos de forma errónea se notificará
Postcondiciones	Se guardarán los nuevos datos del equipo

Tabla 14: CU-AS014 Edición equipo propio

Código	CU-AS015
Caso de uso	Edición jugador
Actor	Usuario
Descripción	El usuario podrá editar cualquier jugador del equipo
Precondiciones	La aplicación habrá sido ejecutada y tendrá mínimo un jugador guardado. El usuario habrá seleccionado un jugador de la lista
Curso normal	1. El usuario ejecuta la aplicación 2. El usuario seleccionará un jugador de la lista 3. El usuario modificará los datos del jugador (nombre, edad, dorsal, posición, goles, tarjetas amarillas*, tarjetas rojas*) 4. El usuario pulsará el botón de guardar
Curso alternativo	
Excepciones	Si el usuario edita los datos de forma errónea se notificará
Postcondiciones	Se guardarán los nuevos datos del jugador

Tabla 15: CU-AS015 Edición jugador

Código	CU-AS016
Caso de uso	Edición equipo rival
Actor	Usuario
Descripción	El usuario podrá editar cualquier equipo rival
Precondiciones	La aplicación habrá sido ejecutada y tendrá mínimo un equipo rival guardado. El usuario habrá seleccionado un equipo de la lista
Curso normal	1. El usuario ejecuta la aplicación 2. El usuario seleccionará un equipo de la lista 3. El usuario modificará los datos del equipo rival (color camiseta, nombre, delegado/entrenador y teléfono) 4. El usuario pulsará el botón de guardar
Curso alternativo	
Excepciones	Si el usuario edita los datos de forma errónea se notificará
Postcondiciones	Se guardarán los nuevos datos del equipo rival

Tabla 16: CU-AS016 Edición equipo rival

Código	CU-AS017
Caso de uso	Edición jornada
Actor	Usuario
Descripción	El usuario podrá editar cualquier jornada
Precondiciones	La aplicación habrá sido ejecutada y tendrá mínimo una jornada guardada. El usuario habrá seleccionado una jornada de la lista
Curso normal	1. El usuario ejecuta la aplicación 2. El usuario irá a la pestaña de “CALENDARIO” 3. El usuario seleccionará una jornada de la lista 4. El usuario modificará los datos de la jornada (equipo local, equipo visitante, fecha, hora, lugar y resultado) 5. El usuario pulsará el botón de guardar
Curso alternativo	
Excepciones	Si el usuario edita los datos de forma errónea se notificará
Postcondiciones	Se guardarán los nuevos datos de la jornada

Tabla 17: CU-AS017 Edición jornada

Código	CU-AS018
Caso de uso	Edición página web
Actor	Usuario
Descripción	El usuario podrá editar la url de la página web
Precondiciones	La aplicación habrá sido ejecutada y tendrá una url introducida. El usuario habrá seleccionado el botón de visualización de página web
Curso normal	1. El usuario ejecuta la aplicación 2. El usuario pulsará el botón de página web 3. El usuario pulsará el botón de editar página web 4. Aparecerá un campo en la parte inferior donde el usuario podrá insertar la nueva url 5. El usuario pulsará el botón de guardar
Curso alternativo	
Excepciones	Si el usuario edita los datos de forma errónea se notificará
Postcondiciones	Se guardará la nueva url

Tabla 18: CU-AS018 Edición página web

Código	CU-AS019
Caso de uso	Borrar jornada
Actor	Usuario
Descripción	El usuario podrá borrar una jornada
Precondiciones	La aplicación habrá sido ejecutada y tendrá mínimo una jornada guardada
Curso normal	1. El usuario ejecuta la aplicación 2. El usuario irá a la pestaña “CALENDARIO” 3. El usuario realizará una pulsación larga sobre la jornada que desea borrar 4. Aparecerá un nuevo botón en la jornada seleccionada 5. El usuario pulsará el botón de borrado 6. Se mostrará el listado de jornadas actualizado
Curso alternativo	1. Si al borrar la jornada, el listado se queda vacío se mostrará el botón de añadir jornada en el centro de la pantalla
Excepciones	Si se produce algún error al borrar la jornada se notificará
Postcondiciones	Se visualizará el listado de jornadas actualizado

Tabla 19: CU-AS019 Borrar jornada

Código	CU-AS020
Caso de uso	Borrar jugador
Actor	Usuario
Descripción	El usuario podrá borrar un jugador
Precondiciones	La aplicación habrá sido ejecutada y tendrá mínimo un jugador guardado
Curso normal	1. El usuario ejecuta la aplicación 2. El usuario irá a la pestaña “EQUIPO” 3. El usuario realizará una pulsación larga sobre el jugador que desee borrar 4. Aparecerá un nuevo botón en el jugador seleccionado 5. El usuario pulsará el botón de borrado 6. Se mostrará el listado de jugadores actualizado
Curso alternativo	1. Si al borrar el jugador, el listado se queda vacío se mostrará el botón de añadir jugador en el centro de la pantalla
Excepciones	Si se produce algún error al borrar el jugador se notificará
Postcondiciones	Se visualizará el listado de jugadores actualizado

Tabla 20: CU-AS020 Borrar jugador

Código	CU-AS021
Caso de uso	Borrar equipo
Actor	Usuario
Descripción	El usuario podrá borrar un equipo rival
Precondiciones	La aplicación habrá sido ejecutada y tendrá mínimo un equipo guardado
Curso normal	1. El usuario ejecuta la aplicación 2. El usuario irá a la pestaña “EQUIPOS” 3. El usuario realizará una pulsación larga sobre el equipo que desee borrar 4. Aparecerá un nuevo botón en el equipo seleccionado 5. El usuario pulsará el botón de borrado 6. Se mostrará el listado de equipos actualizado
Curso alternativo	1. Si al borrar el equipo, el listado se queda vacío se mostrará el botón de añadir equipo en el centro de la pantalla
Excepciones	Si se produce algún error al borrar el equipo se notificará
Postcondiciones	Se visualizará el listado de equipos actualizado

Tabla 21: CU-AS021 Borrar equipo

Código	CU-AS022
Caso de uso	Llamada delegado equipo
Actor	Usuario
Descripción	El usuario podrá realizar una llamada telefónica a cada delegado de los equipos rivales
Precondiciones	La aplicación habrá sido ejecutada y tendrá mínimo un equipo guardado
Curso normal	1. El usuario ejecuta la aplicación 2. El usuario irá a la pestaña “EQUIPOS” 3. El usuario seleccionará un equipo del listado 4. Aparecerá un nuevo botón en la pantalla de edición del equipo 5. El usuario pulsará el botón de llamada 6. Se realizará la llamada telefónica directamente
Curso alternativo	1. Si el equipo no tiene un teléfono asignado se notificará al usuario
Postcondiciones	Se realizará llamada telefónica

Tabla 22: CU-AS022 Llamada delegado equipo

3.2.2 Aplicación wearable

A continuación se listan los casos de uso que interactúan directamente con el usuario en la aplicación del wearable:

- CU-AW001: Visualización jornadas.
- CU-AW002: Visualización jugadores.
- CU-AW003: Visualización equipos rivales.
- CU-AW004: Visualización jornada.
- CU-AW005: Visualización jugador.
- CU-AW006: Visualización equipo.

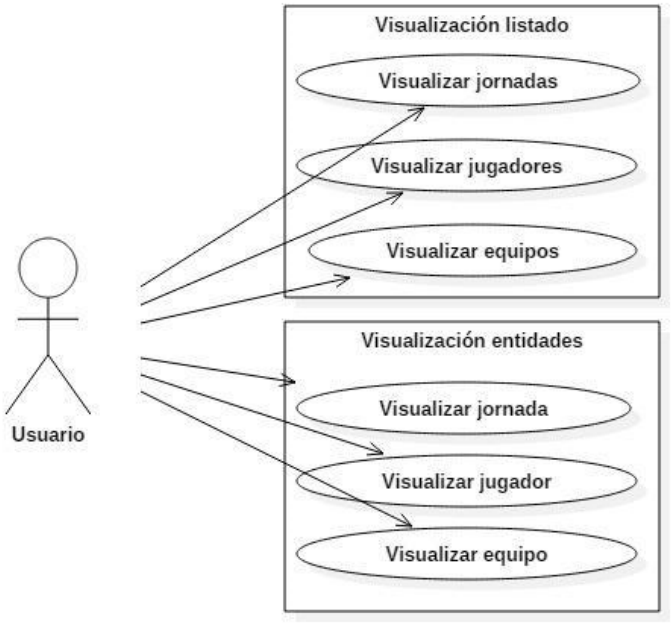

Figura 14: Casos de uso aplicación wear

Una vez identificados los casos de uso para la aplicación del wearable se procede a detallar cada uno de ellos:

Código	CU-AW001
Caso de uso	Visualización jornadas
Actor	Usuario
Descripción	El usuario podrá visualizar en el Smartwatch el listado de las jornadas con su estado
Precondiciones	La aplicación tendrá mínimo una jornada guardada
Curso normal	1. El usuario ejecuta la aplicación en el smartwarch 2. El usuario seleccionará el primero elemento de la lista “CALENDARIO” 3. Se visualizará el listado con las jornadas
Curso alternativo	1. Si no existen jornadas, se notificará al usuario
Excepciones	
Postcondiciones	

Tabla 23: CU-AW001 Visualización jornadas

Código	CU-AW002
Caso de uso	Visualización jugadores
Actor	Usuario
Descripción	El usuario podrá visualizar en el Smartwatch el listado de los jugadores y su nombre
Precondiciones	La aplicación tendrá mínimo un jugador guardado
Curso normal	1. El usuario ejecuta la aplicación en el smartwarch 2. El usuario seleccionará el segundo elemento de la lista “JUGADORES” 3. Se visualizará el listado con los jugadores del equipo
Curso alternativo	1. Si no existen jugadores, se notificará al usuario
Excepciones	
Postcondiciones	

Tabla 24: CU-AW002 Visualización jugadores

Código	CU-AW003
Caso de uso	Visualización equipos rivales
Actor	Usuario
Descripción	El usuario podrá visualizar en el Smartwatch el listado de los equipos rivales y su nombre
Precondiciones	La aplicación tendrá mínimo un equipo rival guardado
Curso normal	1. El usuario ejecuta la aplicación en el Smartwatch 2. El usuario seleccionará el segundo elemento de la lista “EQUIPOS” 3. Se visualizará el listado con los equipos rivales
Curso alternativo	1. Si no existen equipos, se notificará al usuario
Excepciones	
Postcondiciones	

Tabla 25: CU-AW003 Visualización equipos rivales

Código	CU-AW004
Caso de uso	Visualización jornada
Actor	Usuario
Descripción	El usuario podrá visualizar en el Smartwatch el detalle de cada jornada
Precondiciones	La aplicación tendrá mínimo una jornada guardad
Curso normal	1. El usuario ejecuta la aplicación en el Smartwatch 2. El usuario seleccionará el primer elemento de la lista “CALENDARIO” 3. Se visualizará el listado con las jornadas 4. El usuario seleccionará la jornada que quiere visualizar 5. Se mostrará el detalle de la jornada(equipo local, equipo visitante, fecha, hora, lugar y resultado
Curso alternativo	
Excepciones	
Postcondiciones	

Tabla 26: CU-AW004 Visualización jornada

Código	CU-AW005
Caso de uso	Visualización jugador
Actor	Usuario
Descripción	El usuario podrá visualizar en el Smartwatch el detalle de cada jugador del equipo
Precondiciones	La aplicación tendrá mínimo un jugador guardado
Curso normal	1. El usuario ejecuta la aplicación en el Smartwatch 2. El usuario seleccionará el segundo elemento de la lista “JUGADORES” 3. Se visualizará el listado con los jugadores 4. El usuario seleccionará el jugador que quiere visualizar 5. Se mostrará el detalle del jugador(posición, nombre, goles/puntos, edad y dorsal)
Curso alternativo	
Excepciones	
Postcondiciones	

Tabla 27: CU-AW005 Visualización jugador

Código	CU-AW006
Caso de uso	Visualización equipo rival
Actor	Usuario
Descripción	El usuario podrá visualizar en el Smartwatch el detalle de cada equipo rival
Precondiciones	La aplicación tendrá mínimo un equipo rival guardado
Curso normal	1. El usuario ejecuta la aplicación en el Smartwatch 2. El usuario seleccionará el segundo elemento de la lista “EQUIPOS” 3. Se visualizará el listado con los equipos 4. El usuario seleccionará el equipo que quiere visualizar 5. Se mostrará el detalle del equipo (camiseta, nombre equipo, delegado y teléfono)
Curso alternativo	
Excepciones	
Postcondiciones	

Tabla 28: CU-AW006 Visualización equipo rival

4. Diseño de la aplicación

A continuación se detalla la arquitectura de la aplicación, se dividirá en tres partes que son la Base de datos, la aplicación Smartphone y la aplicación Wear.

4.1 Descripción detallada

4.1.1 Base de datos

La aplicación al ejecutarse por primera vez generará una base de datos en el Smartphone, la base de datos consta de 4 tablas, 3 de ellas se utilizarán para persistir los datos de las 3 entidades que componen la aplicación (Jornada, Jugador, Equipo), la restante se utilizará para la configuración de la aplicación, contendrá varios datos de configuración (URL de la página web a cargar, nombre del equipo, color de la camiseta y tipo de deporte).

El modelo Entidad/Relación quedaría de la siguiente manera:

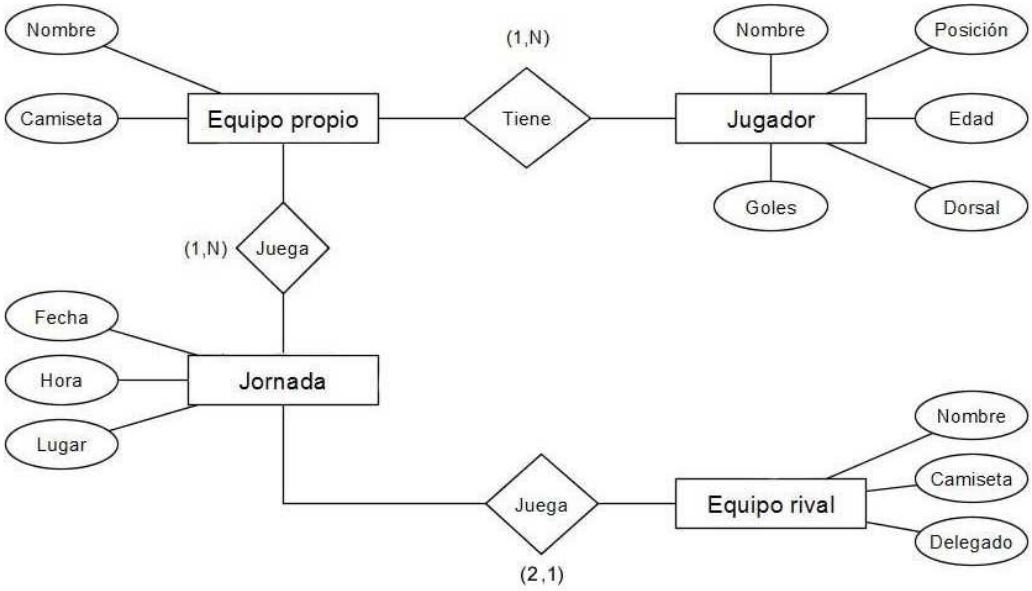

Figura 15: Modelo Entidad-Relación

Detallando el modelo E/R observamos que una Jornada contiene un equipo rival y el propio equipo del usuario que se guardará en la entidad **Config**, mediante código se restringirá a que una jornada no pueda tener dos equipos iguales o que ningún equipo sea el del usuario por lo que siempre habrá una relación “uno a uno” entre **Jornada**-**Equipo** y **Jornada**-**Config**.

A continuación se muestra el detalle de las tablas que componen la base de datos:

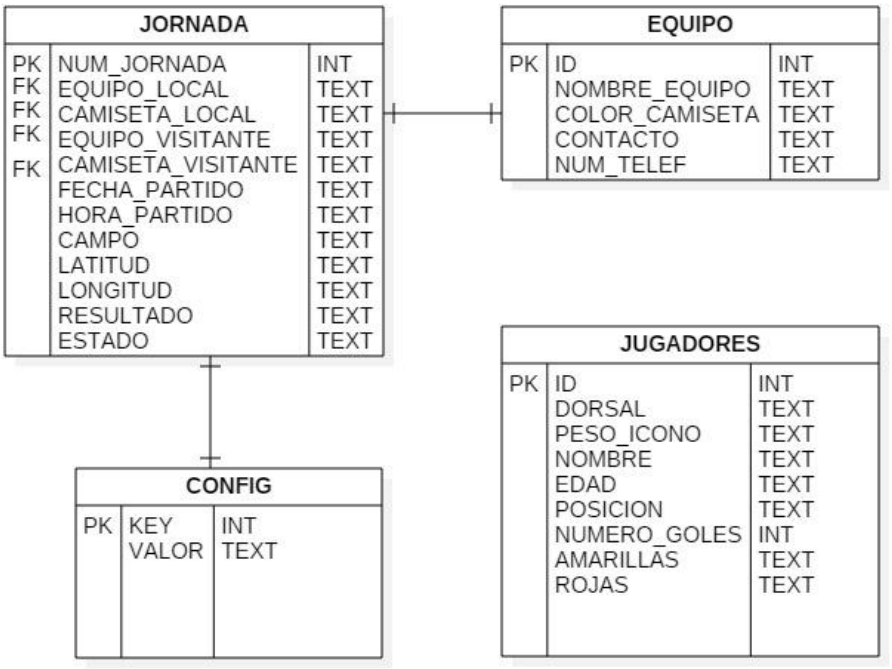

Figura 16: Paso a tablas del modelo Entidad-Relación

Se toma la decisión de no relacionar las entidades en BBDD y realizar la relación a nivel de código, pudiendo insertar en cada jornada un equipo rival (Tabla **EQUIPO**) y el propio equipo de la aplicación persistido en la tabla **CONFIG** (“equipo”, “camiseta”).

4.1.2 Aplicación Smartphone

Para la aplicación móvil se toma la decisión de dividir la implementación mediante paquetes divididos por utilidades:

- **Dao:** En este paquete se añadirán los accesos a los objetos de BBDD, en ellos se creará los distintos métodos para realizar las distintas operaciones con los objetos cómo inserción, borrado, actualización, selección única, selección en listado.
- **Entities:** En este paquete se añadirán las clases encargadas de mapear cada entidad de BBDD con su Objeto.
- **Tabs:** En este paquete se añadirán los “**Fragment**” de la aplicación que controlarán las entidades de la aplicación.

- **Viewers:** Este paquete contiene los "**Activity**" encargados de gestionar la visualización de cada entidad y su posterior edición.
- **Wizards:** Este paquete contiene los "**Activity**" encargados de guiar al usuario al añadir cada nueva entidad.
- **Transfer:** Este paquete ha sido creado para gestionar la conexión entre Smartphone-wearable y la sincronización de datos requeridos por el wearable.
- **Otros:** También se crean paquetes auxiliares para gestionar las excepciones de la aplicación, los adaptadores necesarios para los listados de entidades y enumeradores para gestionar las imágenes a mostrar en cada elemento.

A continuación se detalla cada uno de los paquetes y las clases que los componen:

DAOs: En el paquete *team.jose.com.team.dao* se incluyen las clases que conforman el modelo de la aplicación:

- **BaseDatos**: Esta clase extiende de *SQLiteOpenHelper* y permite crear una base de datos interna en el propio terminal la cual se utilizará para persistir la información de las entidades que la componen, en esta clase se definen las tablas y sus campos:

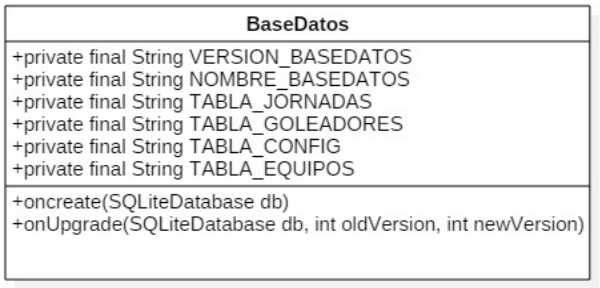

Figura 17: Detalle de clase BaseDatos

- **ConfigDao, EquipoDao, JornadaDao y JugadorDao**: Estas clases actúan como acceso a la base de datos, realizan la conexión mediante "dataBase.getWritableDatabase()" y ejecutan la sentencia necesaria para cada acción solicitada por el usuario como por ejemplo, obtener datos de un jugador, ver listado de jornadas, actualizar equipo rival etc...

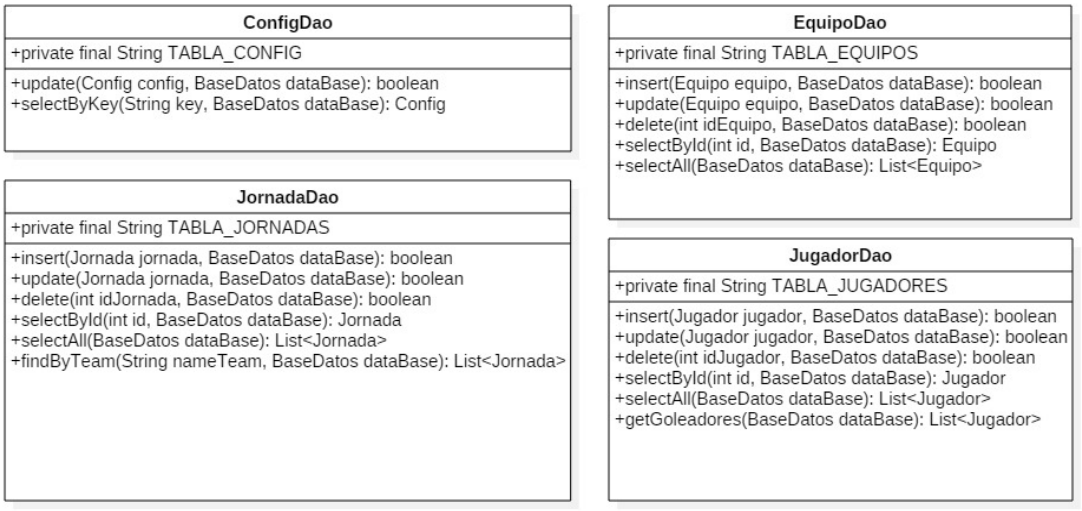

Figura 18: Detalle clases Daos

- **Entities:** En el paquete *team.jose.com.team.entities* se incluyen las entidades de la aplicación para realizar la conversión del modelo de Base de datos con el código Java. Existe una clase entidad por cada una de las tablas de la base de datos. Estas clases son **Config**, **Equipo**, **Jornada** y **Jugador.**

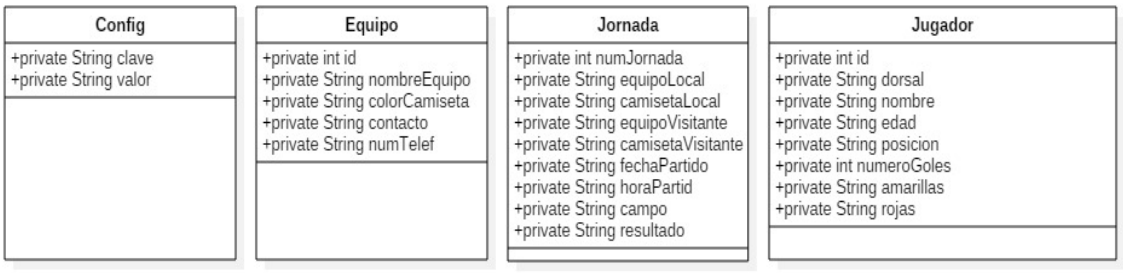

Figura 19: Detalle clases Entidades

- **Tabs:** El paquete *team.jose.com.team.tabs* incluye las cuatro pestañas de la pantalla principal **Calendario**, **Equipo**, **Equipos**, **Goleadores**, en estas pestañas se muestran los listados de las entidades descritas anteriormente.

Estas clases extienden de **Fragment** lo que significa que con una sola interfaz se controlan los 4 listados independientemente, a continuación se describe una de ellas solamente debido al gran parecido entre ellas.

La estructura es la siguiente:

onCreateView, este método es llamado cada vez que el usuario decide visualizar la pestaña. Se hace una consulta a base de datos para mostrar el listado de la entidad requerida, por ejemplo el listado de Jornadas. Se inicializan todos los componentes de la interfaz que maneja este **Fragment**, como el botón de añadir una nueva entidad(**addButton.setOnClickListener**), el listener que permite borrar un elemento de la lista mediante una pulsación prolongada(**list.setOnItemLongClickListener)** o la visualización de cada entidad mediante una pulsación simple(**list.setOnItemClickListener**).

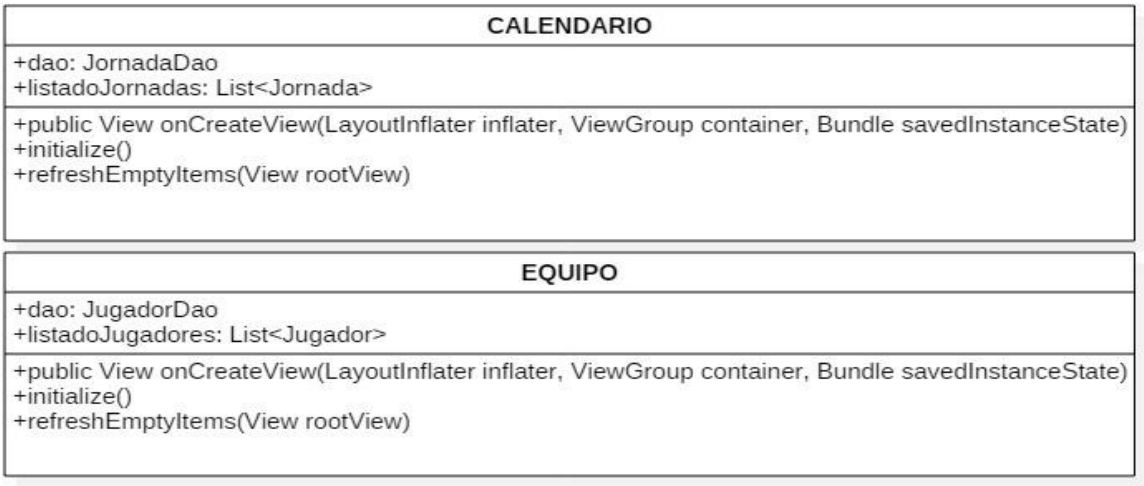

Figura 20: Detalle clases Tabs

Como ocurre con los **tabs** anteriores estas clases son similares entre sí, por lo que se detalla únicamente una de ellas (**VerJornada**):

- Contiene la entidad que se visualizará y su dao (**Jornada**, **JornadaDao**), necesarios para obtener la información de la jornada seleccionada de la base de datos y mostrarla al usuario.
- Elementos de la interfaz gráfica, como pueden ser **TextView**, **EditText**, **ImageView**, **Spinner**...estos elementos se utilizan para mostrar la información de la jornada de forma estructurada y permitir al usuario modificar esa información para una posterior actualización si así se desea.
- **onCreate,** este método será el encargado de inicializar la interfaz de visualización de la jornada y rellenar todos los datos de la jornada. En el método **initialize(int idJornada)** se llamará a base de datos para obtener la información e inicializar todos los elementos descritos en el punto anterior, esta inicialización se hace

referenciando cada elemento de la interfaz definida en su fichero xml correspondiente.

- **check(),** este método será el encargado de comprobar si toda la información modificada por el usuario es correcta antes de actualizar la Jornada en base de datos, de no ser así se mostrará un mensaje de error al usuario mediante el método **showError(String text).**
- **onKeyDown(int keyCode, KeyEvent event**), todas las **Activity** de este tipo viewer tendrán este método, el cual detecta la pulsación del botón atrás en el dispositivo y regresará al usuario a la pantalla principal si no desea realizar ninguna actualización.

Figura 21: Detalle clases VerJornada

Wizards: El paquete *team.jose.com.team.wizards* contiene los Activity **NuevaJornada**, **NuevoEquipo** y **NuevoJogador** que permiten al usuario crear las entidades **Jornada**, **Jugador** y **Equipo,** se detalla a continuación la encargada de añadir un nuevo Jugador:

- Este **Activity** contiene los elementos de la interfaz que el usuario deberá rellenar para añadir el nuevo Jugador, estos elementos son: **nombreJugador**(EditText), **dorsal**(EditText), **edad**(EditText) y cada uno de los Button que representan las posibles **posiciones** en el equipo (portero, defensa, medio...), para el caso de seleccionar un equipo de baloncesto estas posiciones variarán (base, escolta, alero…).
- **onCreate,** este método, como ocurre con todas las **Activity** descritas es el encargado de llamar al método initialize() para inicializar todos los elementos de la interfaz referenciándolos con el ficheros xml correspondiente. Se dotará al botón guardar de la interfaz el "escuchador" **guardar.setOnClickListener** que detectará la pulsación del usuario para realizar el guardado de la

entidad si todos los datos introducidos por el usuario son correctos, de no ser así se notificará al usuario.

- **onKeyDown(int keyCode, KeyEvent event),** estas **Activity** también detectarán la pulsación del usuario si quisiera cancelar la operación de añadir y regresar a la pantalla principal.

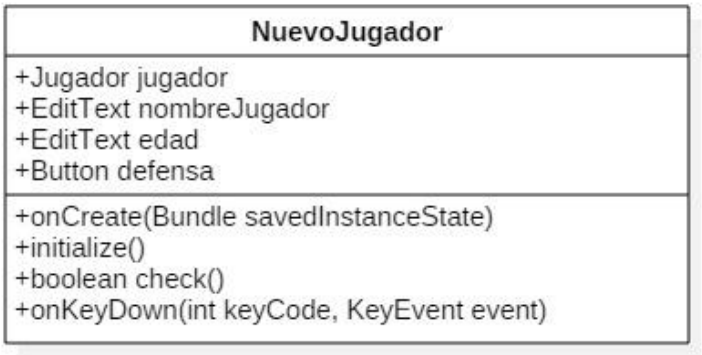

Figura 22: Detalle clase NuevoJugador

Transfer: Este paquete *team.jose.com.team.wizards* contiene las clases **SendToDataLayerThread**, **MessageReceiver** y **ListenerService**, que serán las encargadas de realizar la conexión con el módulo de la aplicación wear y transferir la información:

- **SendToDataLayerThread,** esta clase extiende de Thread y comprobará si existen "nodos" activos, conectados con el dispositivos móvil, entendemos como nodos cualquier dispositivo conectado mediante bluetooth, de ser así se transferirá la información solicitada por el dispositivo wear mediante la instrucción **Wearable.MessageApi.sendMessage.**
- **MessageReceiver,** esta clase extiende de BroadcastReceiver y será la encargada de inicializar un objeto del tipo GoogleApiClient necesario para transmitir información entre ambos dispositivos, una vez inicializado el objeto se realizará la conexión con el dispositivo wear y se enviará la información utilizando la clase descrita anteriormente.
- **ListenerService,** esta clase extiende de **WearableListenerService,** la cual se declara como un servicio Android, se deberá añadir en el fichero AndroidManifest.xml, este servicio será el encargado de estar escuchando constantemente y detectar que se solicita información por parte del Smartwatch, una

vez detectada la llamada se llamará a la clase **MessageReceiver** anterior.

- La clase **SendToDataLayerThread** al transferir la información al Smartwatch se apoya en la clase auxiliar **GestionMensajes**, que será la encargada de analizar el mensaje recibido desde el Smartwatch y dependiendo del tipo de mensaje enviará una información u otra.

Existen otros paquetes auxiliares como *team.jose.com.team.enums* y *team.jose.com.team.exception* que contendrán clases auxiliares como **Colores**, **Posiciones** y **EstadoJornada** que guardarán una relación entre el nombre y la imagen a mostrar, se aclara en un ejemplo:

- Para identificar la camiseta a mostrar para un equipo en concreto, se obtiene el color de la camiseta almacenada en base de datos y utilizando la clase Colores se sabrá qué imagen mostrar:
  - Blanco ("Blanco", R.drawable.cam_blanca).
  - Azul ("Azul", R.drawable.cam_azul).

El paquete *team.jose.com.team.exception* se ha utilizado para crear una excepción personalizada que extienda de **Exception**, esta excepción se ha creado para personalizar las excepciones devueltas por los accesos a la base de datos o para controlar las posibles excepciones inesperadas y mostrar al usuario un mensaje “amigable”.

4.1.3 Aplicación Wear

La estructura de la aplicación wear es similar a la de la aplicación móvil por lo que no se volverá a detallar en profundidad, cabe destacar que algunas clases definidas en la aplicación móvil no se incluyen en la aplicación wear como pueden ser **NuevoJugador**, **NuevaJornada**, **NuevoEquipo** ya que la parte wear simplemente incluye las funciones de visualizar entidades y no de añadir o editarlas.

4.2 Diagramas de clase

A continuación se detalla la estructura de la aplicación para una rápida y fácil comprensión de todos los elementos que la componen:

4.2.1 Aplicación Móvil

En este apartado se detalla la estructura de la aplicación móvil, para una fácil comprensión se añade un esquema de clases, las cuales se detallan una por una más adelante.

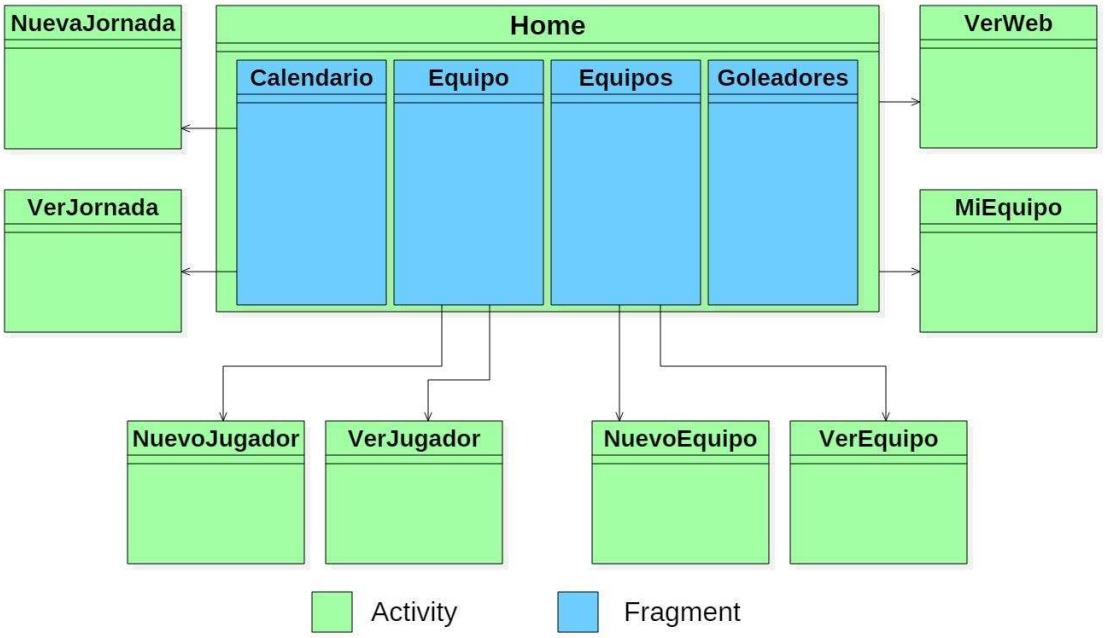

Figura 23: Diagrama de clases de la aplicación móvil

La aplicación móvil se compone por una clase principal **Home**, la cual contiene cuatro Fragments (Calendario, Equipo, Equipos y Goleadores), cada uno de estos Fragments controlará los listados de las entidades.

- **Home**, esta clase principal además de gestionar los Fragments también podrá lanzar dos actividades adicionales como son **NuevaWeb** que visualizará la página web elegida por el usuario y la actividad **MiEquipo** donde se podrá modificar el nombre y el color de la camiseta del equipo del usuario.

- **Calendario,** este Fragment podrá lanzar dos actividades como son **NuevaJornada** y **VerJornada**, encargadas de añadir, visualizar y editar una jornada.
- El Fragment **Equipo** podrá lanzar las actividades **NuevoJugador** y **VerJugador,** encargadas de añadir, visualizar y editar un jugador.
- El Fragment Equipos podrá lanzar las actividades **NuevoEquipo** y **VerEquipo** que se encargarán de añadir, visualizar y editar los equipos rivales.
- **Goleadores**, este Fragment simplemente visualizará la lista de Goleadores/Anotadores del equipo.

4.2.2 Aplicación Wear

En este apartado se detalla la estructura de la aplicación wear, para una fácil comprensión se añade un esquema de clases, las cuales se detallan más adelante.

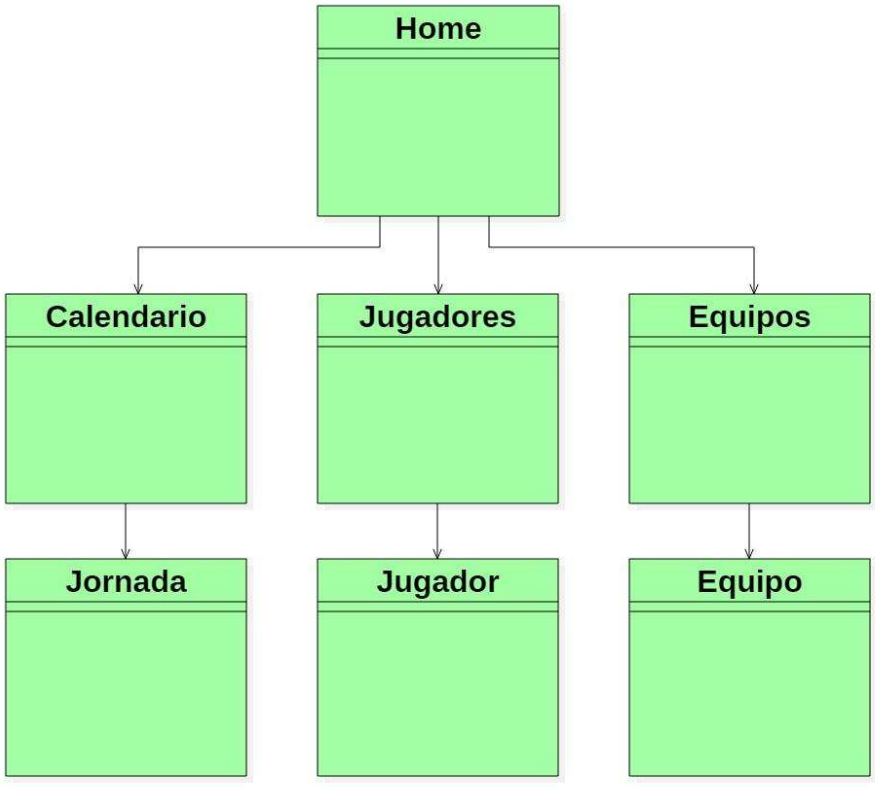

Figura 24: Diagrama de clases de la aplicación Wear

La aplicación wearable parte de una clase principal **Home** donde se mostrará un listado de tres elementos que son **Calendario**, **Jugadores** y **Equipos**, una vez seleccionado cualquiera de los tres elementos del listado se lanzará su actividad correspondiente, por ejemplo, seleccionando el elemento **Calendario** se accede al listado de Jornadas en el caso de que hubiera alguna añadida. Una vez dentro del listado elegido se podrá seleccionar uno de los elementos y se lanzará la actividad correspondiente para visualizar la entidad seleccionada.

4.3 Diagramas de flujo

En este apartado se describen los flujos de las dos partes de la aplicación, primero se detallará la aplicación móvil y después la aplicación wear.

4.3.1 Aplicación Móvil

Para una fácil comprensión de los diagramas de secuencia individuales, se muestra en la siguiente imagen el diagrama “común” de flujo que se realizará entre el usuario y la aplicación móvil:

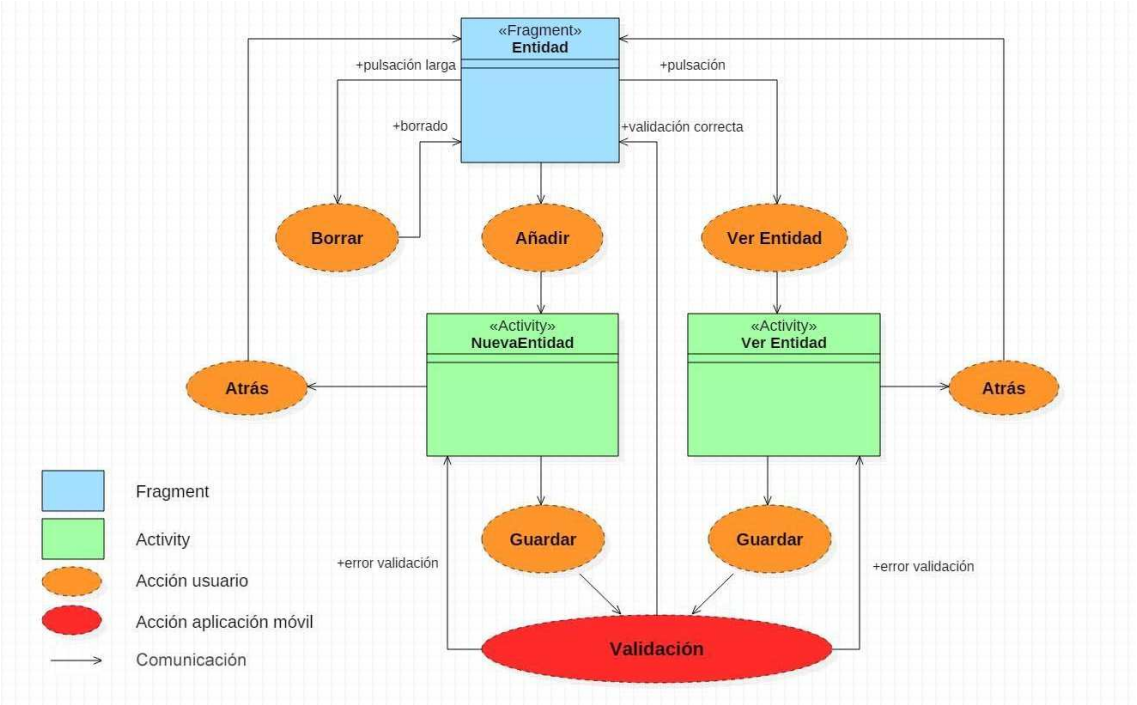

Figura 25: Diagrama de flujo de la aplicación Móvil

El <<Fragment>> corresponde a cada una de las pestañas de la pantalla principal de la aplicación (Calendario, Equipo y Equipos), desde la pestaña que el usuario elija podrá realizar una de las 3 acciones que se indican en la figura (Borrar, Añadir y Ver Entidad), si el usuario pulsa prolongadamente en cualquier elemento del listado que se le muestra (Debe existir al menos un elemento) al usuario se le dará la opción pulsando el botón de borrar, eliminar el elemento seleccionado.

Otra de las acciones que el usuario podrá realizar es añadir una entidad, para realizar esta acción el usuario deberá pulsar el botón de añadir y se le redirigirá a una nueva pantalla <<Activity>> donde podrá rellenar los datos de la entidad que desea añadir. Una vez introducidos los datos y al pulsar el botón de guardar, la aplicación internamente comprobará que los datos son correctos y se retornará al usuario a la pantalla principal donde se visualizará la nueva entidad.

Si el usuario desea visualizar cualquiera de las entidades sólo deberá seleccionarla en el listado correspondiente, la aplicación mostrará la pantalla <<Activity>> con los datos de la entidad seleccionada, una vez en esta pantalla el usuario podrá editar cualquiera de los datos de la entidad. El usuario pulsará el botón de guardar y una vez se haya comprobado por la aplicación que los datos editados son correctos se retornará a la pantalla principal y se visualizará la entidad modificada.

Desde cualquiera de las pantallas de visualizar o añadir nueva entidad, el usuario podrá volver hacia atrás a la pantalla principal sin necesidad de guardar los cambios o sin añadir la entidad.

En cualquiera de los dos casos de añadir o visualizar/editar, si los datos añadidos o editados no son correctos se notificará al usuario los errores y se mantendrá en la pantalla actual en la que se encuentre.

4.3.2 Aplicación Wear

Al igual que ocurre en los flujos de la aplicación móvil, los flujos de la aplicación wearable son muy similares por lo que se detalla solo uno de ellos a modo de ejemplo, el visualizar jugador:

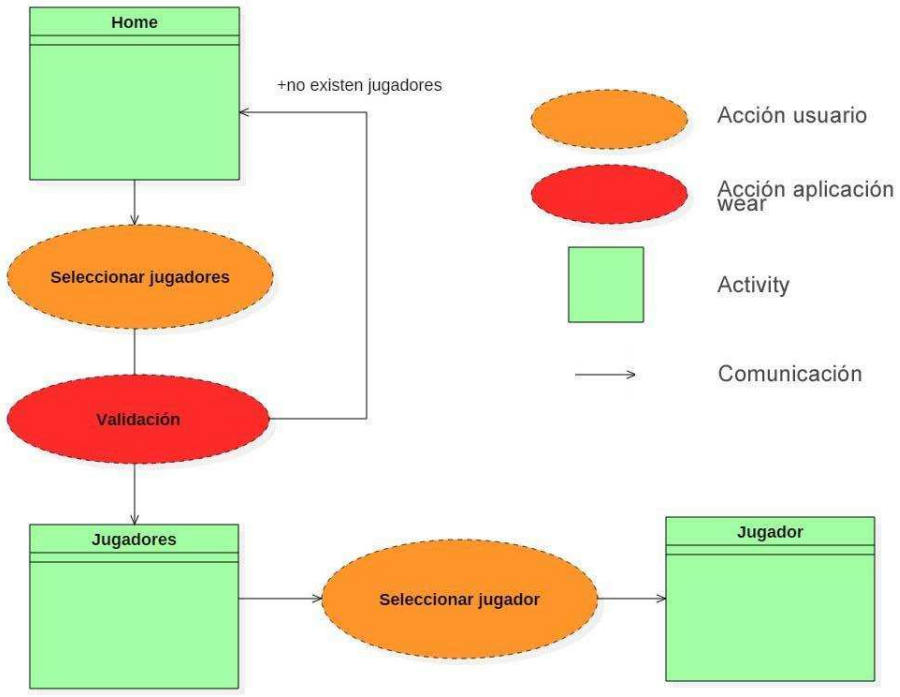

Figura 26: Diagrama de flujo de la aplicación Wear

Lanzada la aplicación en el Smartwatch, se mostrará un listado con tres elementos, **Calendario**, **Jugadores** y **Equipos**, el usuario deberá seleccionar uno de ellos, una vez seleccionado la aplicación wear se comunicará con la aplicación móvil solicitando la información requerida por el usuario, en el caso que nos ocupa, el usuario decide seleccionar el elemento Jugadores, donde se mostrará el listado con todos los jugadores del equipo, si no existiera ningún jugador en la aplicación se le notificará al usuario y la aplicación se mantendrá en la actividad principal. Si existe algún jugador se lanzará la actividad **Jugadores** donde se muestra el listado, en este punto el usuario deberá seleccionar un jugador y se lanzará la siguiente actividad **Jugador** dónde se visualizará los datos del jugador seleccionado.

El flujo es el mismo tanto si se selecciona **Calendario** como **Equipos**, lanzándose la actividad correspondiente con el listado de la entidad seleccionada y posteriormente se lanzará la actividad que muestra la información individual de cada entidad seleccionada en el listado.

4.4 Diagramas de secuencia

En este apartado se detalla los diagramas de secuencia de cada una de las posibles acciones que el usuario podrá realizar sobre la aplicación. Se dividirán los diagramas de secuencia para una fácil comprensión en dos partes, la parte de aplicación Smartphone y la aplicación wearable.

4.4.1 Aplicación Smartphone

En la parte de la aplicación Smartphone se darán las siguientes secuencias que describirán las acciones que realizará el usuario.

- **Añadir jornada:**

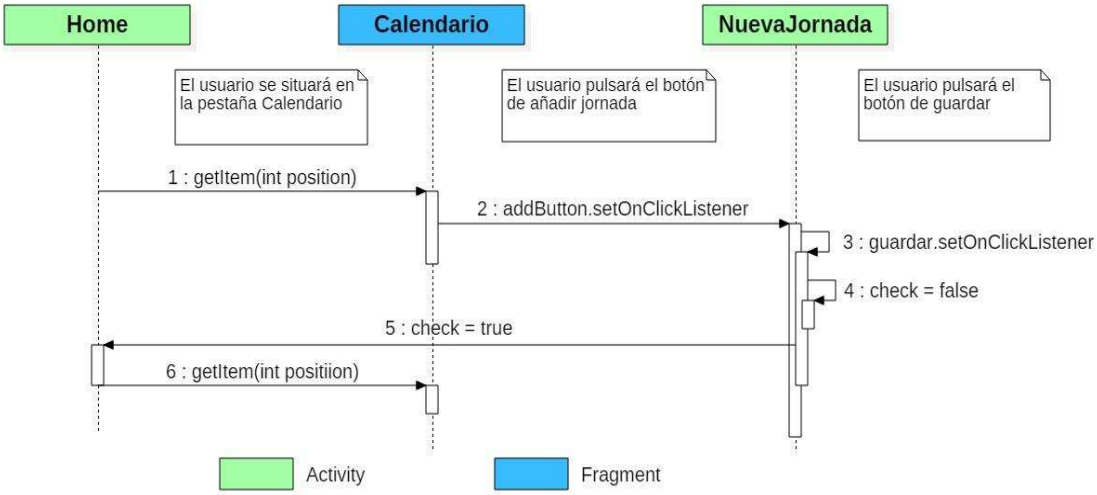

Figura 27: Diagrama de secuencia añadir jornada

1. El usuario una vez dentro de la aplicación se situará en la primera pestaña **Calendario**.
2. El usuario pulsará el botón de añadir jornada y el <<Fragment>> lanzará mediante la instrucción startActivity lanzará el Activity **NuevaJornada**.
3. El usuario una vez en la pantalla de **NuevaJornada** rellenará los datos de la jornada y pulsará el botón guardar.

4. El Activity **NuevaJornada** verificará que los datos introducidos por el usuario son correctos, de no ser así la aplicación se mantendrá en el mismo Activity y notificará al usuario los datos erróneos.
5. El Activity **NuevaJornada** verificará que los datos introducidos por el usuario son correctos, si lo fueran la aplicación lanzaría de nuevo el Activity **Home**.
6. **Home** reconocerá mediante un parámetro recibido que debe mostrar la primera pestaña **Calendario**.

- **Añadir Jugador**:

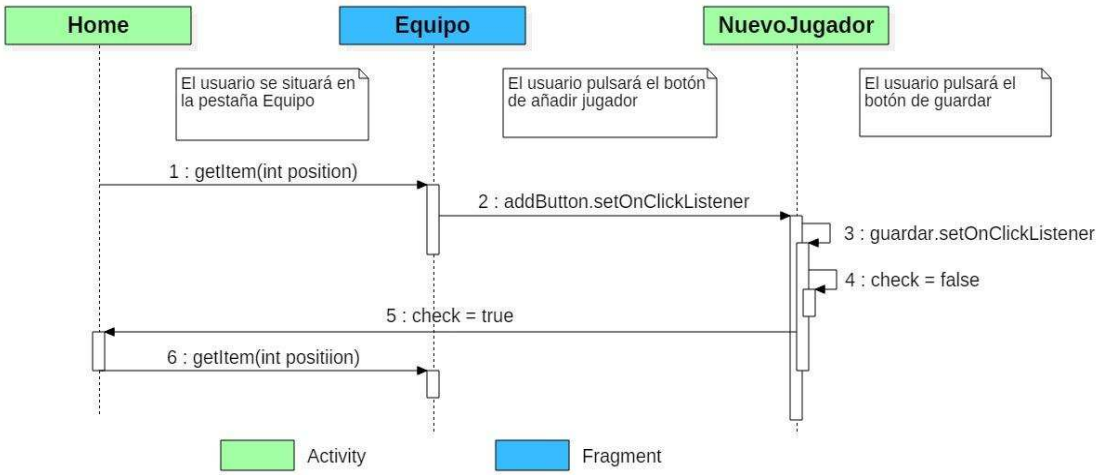

Figura 28: Diagrama de secuencia añadir jugador

1. El usuario una vez dentro de la aplicación se situará en la segunda pestaña **Equipo**.
2. El usuario pulsará el botón de añadir jugador y el <<Fragment>> lanzará mediante la instrucción startActivity lanzará el Activity **NuevoJugador**.
3. El usuario una vez en la pantalla de **NuevoJugador** rellenará los datos del jugador y pulsará el botón guardar.
4. El Activity **NuevoJugador** verificará que los datos introducidos por el usuario son correctos, de no ser así la aplicación se mantendrá en el mismo Activity y notificará al usuario los datos erróneos.
5. El Activity **NuevoJugador** verificará que los datos introducidos por el usuario son correctos, si lo fueran la aplicación lanzaría de nuevo el Activity **Home**.
6. **Home** reconocerá mediante un parámetro recibido que debe mostrar la primera pestaña **Equipo**.

- **Añadir equipo:**

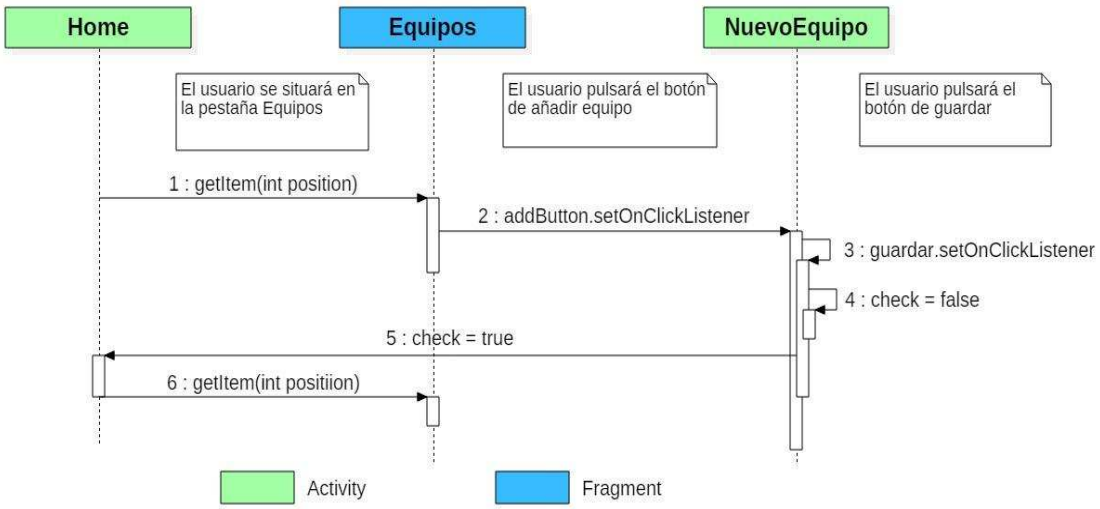

Figura 29: Diagrama de secuencia añadir equipo

1. El usuario una vez dentro de la aplicación se situará en la tercera pestaña **Equipos**.
2. El usuario pulsará el botón de añadir equipo y el <<Fragment>> lanzará mediante la instrucción startActivity lanzará el Activity **NuevoEquipo**.
3. El usuario una vez en la pantalla de **NuevoEquipo** rellenará los datos del equipo y pulsará el botón guardar.
4. El Activity **NuevoEquipo** verificará que los datos introducidos por el usuario son correctos, de no ser así la aplicación se mantendrá en el mismo Activity y notificará al usuario los datos erróneos.
5. El Activity **NuevoEquipo** verificará que los datos introducidos por el usuario son correctos, si lo fueran la aplicación lanzaría de nuevo el Activity **Home**.
6. **Home** reconocerá mediante un parámetro recibido que debe mostrar la primera pestaña **Equipos**.

Para simplificar el número de diagramas de secuencia se utilizará uno “común” para **visualizar/editar entidad** y **eliminar entidad**:

- **Visualizar/Editar entidad**:

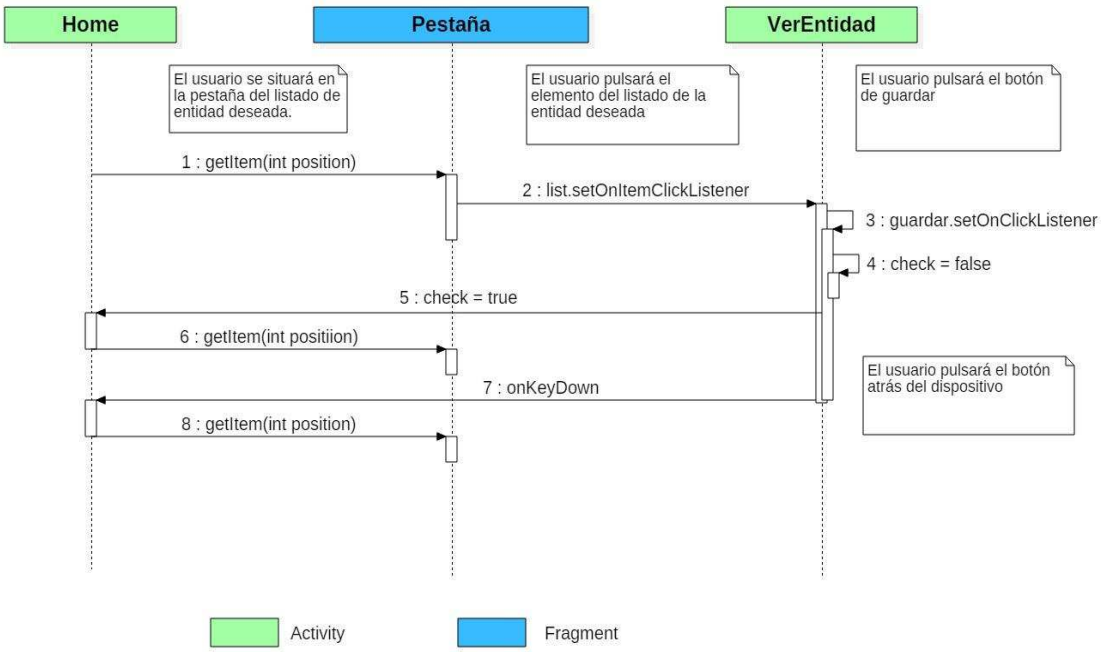

Figura 30: Diagrama de secuencia visualizar/editar entidad

1. El usuario una vez dentro de la aplicación se situará en la pestaña deseada (**Calendario**, **Equipo** o **Equipos**).
2. El usuario pulsará en el elemento del listado que desee visualizar o editar, el <<Fragment>> mediante un escuchador en el listado comprobará que elemento ha sido seleccionado y lanzará mediante la instrucción startActivity() el Activity **VerEntidad correspondiente (VerJornada, VerJugador o VerEquipo)**.
3. El usuario una vez en la pantalla de **VerEntidad** rellenará los datos del equipo y pulsará el botón guardar.
4. El Activity **VerEntidad** verificará que los datos introducidos por el usuario son correctos, de no ser así la aplicación se mantendrá en el mismo Activity y notificará al usuario los datos erróneos.
5. El Activity **VerEntidad** verificará que los datos introducidos por el usuario son correctos, si lo fueran la aplicación lanzaría de nuevo el Activity **Home**.
6. **Home** reconocerá mediante un parámetro recibido que debe mostrar la pestaña desde la que comenzó la secuencia.
7. Si el usuario simplemente desea visualizar la entidad correspondiente, para volver a la pantalla principal deberá pulsar el

botón atrás del dispositivo y el Activity **VerEntidad** retornará a **Home**.

8. **Home** reconocerá mediante un parámetro recibido que debe mostrar la pestaña desde la que comenzó la secuencia.

- **Eliminar entidad**:

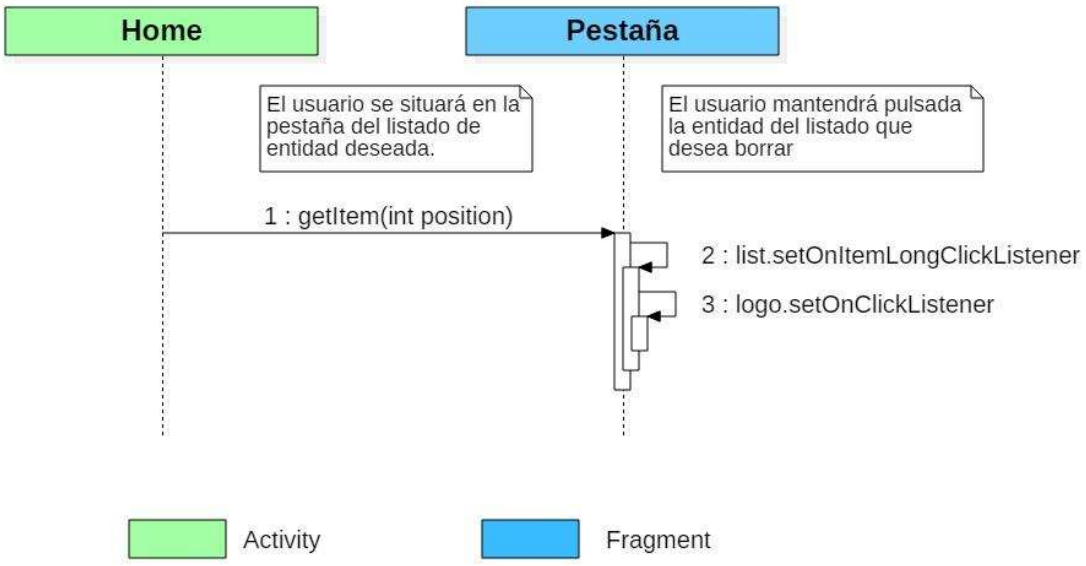

Figura 31: Diagrama de secuencia eliminar entidad

1. El usuario una vez dentro de la aplicación se situará en la pestaña deseada (**Calendario**, **Equipo** o **Equipos**).
2. El usuario mantendrá una pulsación larga en el elemento del listado que desee eliminar, el <<Fragment>> mediante un escuchador en el listado comprobará que elemento ha sido seleccionado y actualizará el elemento mostrando un nuevo botón de borrar.
3. El usuario pulsará sobre el nuevo botón de borrar. Una vez borrada la entidad el listado se refrescará.

- **Visualizar/Editar Web**:

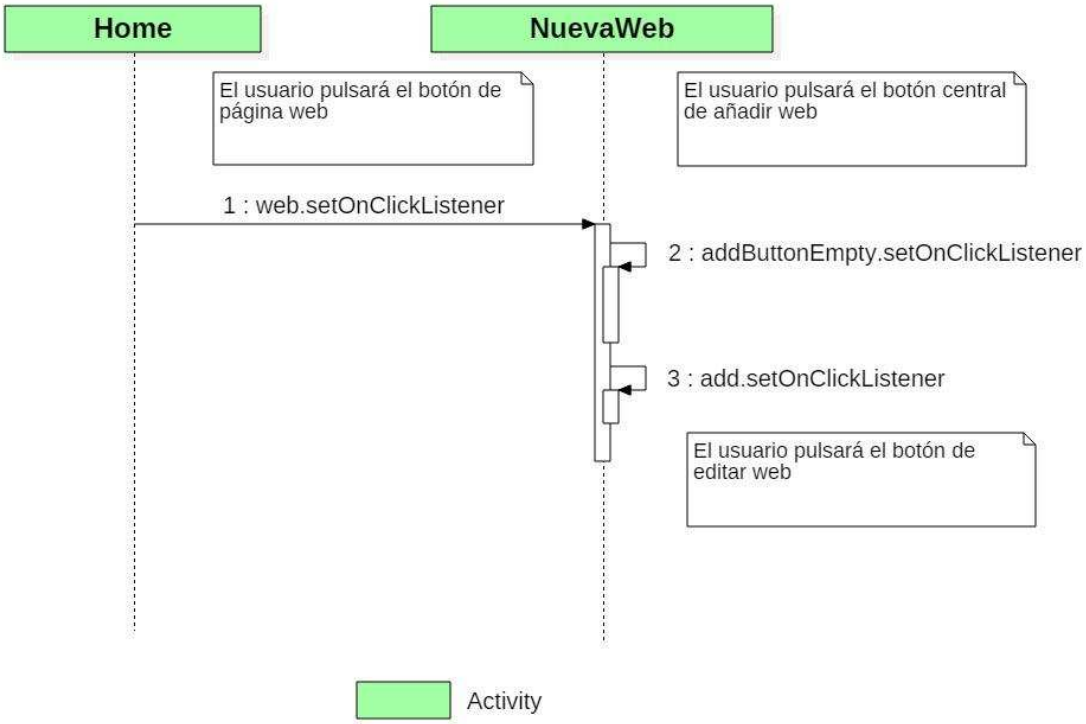

Figura 32: Diagrama de secuencia visualizar/editar web

1. El usuario una vez dentro de la aplicación pulsará el botón situado en la barra superior el botón de página web, el Activity Home lanzará mediante **startActivity()** el Activity **NuevaWeb**.
2. Si no existiera ninguna url añadida en la aplicación la nueva pantalla mostrará un botón central que el usuario deberá pulsar para añadir una nueva url.
3. Cada vez que el usuario desee editar la url deberá pulsar el botón de la barra superior para editarla.

4.4.2 Aplicación Wear

La aplicación wearable estará formada por Activities que contienen y manejan listados de elementos, al pulsar sobre uno de los elementos se mostrará otra nueva pantalla con otro listado, en este segundo paso se controlará la pulsación del elemento y mostrará su detalle. Para simplificar el número de diagramas se mostrará la secuencia en un sólo diagrama común.

Cada vez que se quiera mostrar alguna información en el Smartwatch el dispositivo establece una conexión con el Smartphone el cual procesa la petición y devolverá una respuesta al Smartwatch.

- **Visualizar Entidad**:

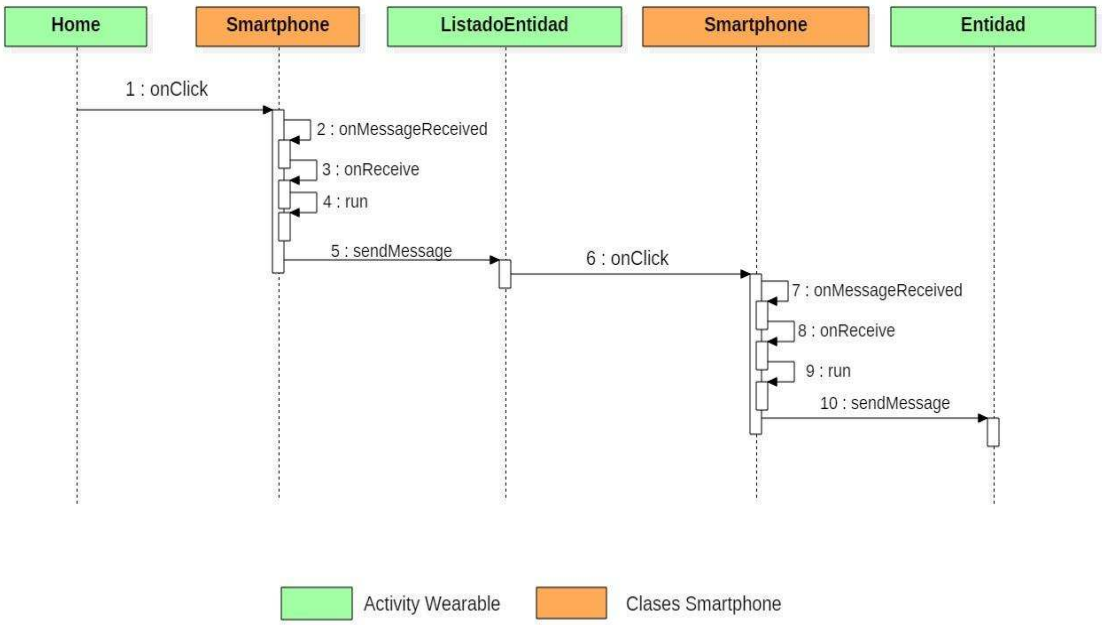

Figura 33: Diagrama de secuencia visualizar entidad aplicación wear

1. El usuario ejecuta la aplicación en el Smartwatch y seleccionará un elemento del listado (Calendario, Jugadores o Equipos). El dispositivo enviará un mensaje al Smartphone.
2. El Smartphone tendrá un servicio en background que escucha los mensajes procedentes del dispositivo conectado mediante bluetooth, una vez detectado el mensaje lanzará el receptor broadcast.

3. El receptor broadcast establecerá la conexión para devolver un mensaje con la información solicitada. A continuación se creará el objeto que enviará la información.
4. Este objeto extiende de Thread, una vez arrancado preparará un mensaje con la información solicitada.
5. Se enviará la información hacia el Smartwatch.
6. Los pasos 6, 7, 8, 9 y 10 son idénticos a los puntos 1, 2, 3, 4 y 5 pero el mensaje intercambiado será diferente solicitando información distinta.

5. Diseño gráfico

5.1 Aplicación Smartphone

En este apartado se detallará la solución tomada en cuanto a diseño de la interfaz de la aplicación, se mostrarán y se detallarán todos los elementos que componen la aplicación en cuanto a visualización ya sea la pantalla principal, listados, visualización de entidades, iconos, etc…

Por lo general se toma la decisión de hacer un diseño simple y sin mucha carga de elementos en la interfaz de usuario. Los iconos se crean con diseños planos y sin sombreados.

Para la selección del deporte una vez se lanza la aplicación por primera vez se decide crear dos iconos, cada uno de ellos representa un jugador de cada deporte (fútbol y baloncesto):

Figura 34: Iconos selección deporte

Para la selección del color de la camiseta se decide mostrar nueve colores de camiseta distintas, diferenciándolas por deporte:

Figura 35: Colores camiseta futbol

Figura 36: Colores camiseta baloncesto

La primera pantalla de la aplicación, iniciada la aplicación por primera vez queda de la siguiente forma:

Figura 37: Pantalla selección deporte

El diseño de la interfaz principal consta de un ActionBar principal y una serie de pestañas donde se podrán visualizar los listados de las entidades, si no existen entidades aparecerán los siguientes iconos para añadir una nueva entidad ya sea una nueva Jornada, un nuevo Jugador o un nuevo Equipo:

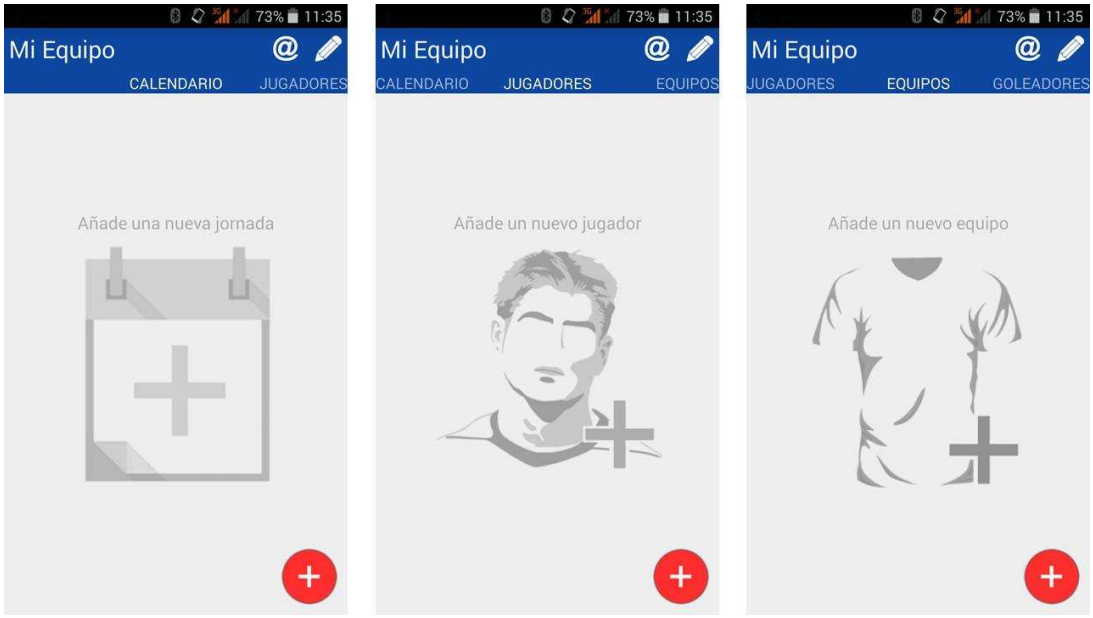

Figura 38: Pantalla principal aplicación móvil

A continuación se detalla el listado de Jornadas de la pestaña de Calendario, en la siguiente figura se mostrará un elemento del, en él aparecerá un icono a la izquierda que mostrará el estado de la jornada:

Figura 39: Estado jornada

El cual indicará si la jornada “no se ha jugado”, “se ha ganado”, “empatado” o “perdido”. En la siguiente figura se muestra el elemento al completo, también se incluye el número de jornada, la fecha y el resultado.

Figura 40: Elemento listado jornadas

En el listado de jugadores, la estructura es similar a la de jornadas, el icono mostrará la posición del jugador, la cual se elegirá cada vez que se añada una nueva posición, se han creado cinco iconos por cada deporte para identificar la posición:

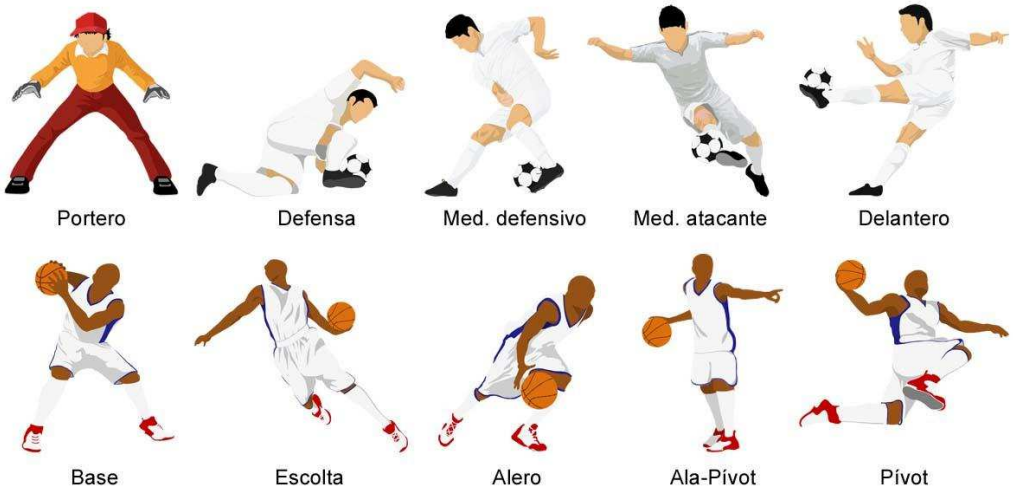

Figura 41: Posiciones deportes

Cada elemento también mostrará el nombre del jugador, la posición y el dorsal:

Figura 42: Elemento listado jugadores

En el listado de equipos el icono de la izquierda mostrará la equipación del equipo, el nombre del equipo y el entrenador:

Figura 43: Elemento listado equipos

En la última pestaña de la página principal se mostrará el listado de los goleadores o anotadores dependiendo del deporte, en cada elemento se mostrará un icono a la izquierda indicando el orden del listado:

Figura 44: Elemento listado goleadores/anotadores

A continuación se detalla la interfaz de añadir entidades, en la siguiente figura se muestra las tres interfaces para añadir cada una de las entidades:

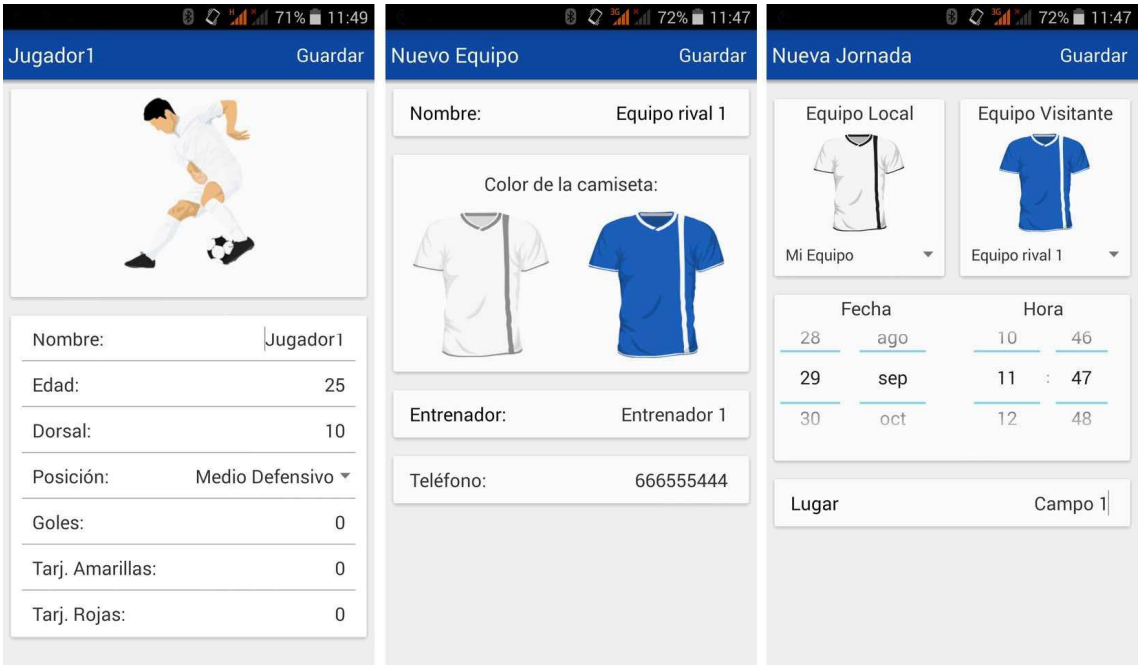

Figura 45: Pantalla añadir entidades

En la primera imagen se muestra la interfaz de añadir jugador, en esta pantalla se muestran el **Nombre**, **Edad**, **Dorsal**, **Posición**, **Goles**, **Tarjetas Amarillas** y **Tarjetas Rojas**…

En la segunda imagen se muestra la interfaz de añadir un nuevo equipo rival, se debe añadir el **Nombre** del equipo de manera obligatoria y seleccionar uno de los iconos de color de la camiseta, los campos **Entrenador** y **Teléfono** son opcionales y el usuario puede dejarlos sin rellenar.

La tercera imagen pertenece a la interfaz de añadir una nueva jornada, se debe seleccionar los equipos **Local** y **Visitante**, la **Fecha** y **Hora** y el **Lugar** del partido.

5.2 Aplicación Wear

La parte de la aplicación wear se compone de listados simples con un icono a la izquierda y el nombre de cada elemento.

En la pantalla principal se muestran tres elementos, como son el calendario, los jugadores y los equipos.

Figura 46: Interfaz principal aplicación wear

Una vez seleccionado cualquiera de los tres elementos se irá a una segunda pantalla mostrando el listado correspondiente al elemento seleccionado, a continuación en la siguiente figura se muestran los listados del Calendario y los Jugadores.

Figura 47: Listados interfaz aplicación wear

Una vez seleccionado el elemento del listado anterior se mostrará el detalle, en este caso y para ilustrar este tipo de pantallas se mostrará el detalle de una jornada y el detalle de un jugador.

Figura 48: Detalle entidades aplicación wear

6. Conclusiones

Este proyecto se ha realizado en base a una idea inicial del desarrollo de una aplicación para dispositivos con sistema operativo Android, donde los usuarios que utilizan estos dispositivos puedan insertar, consultar y modificar información sobre su equipo de fútbol como los jugadores de su propio equipo, el calendario de la temporada, resultados …

Profundizando en el mercado de aplicaciones para dispositivos Android no se encuentran muchas aplicaciones relacionadas con esta temática. Debido al escaso número de aplicaciones se le añade un añadido y se hace compatible con dispositivos wearables pudiendo consultar la información en la muñeca del usuario.

Se decide elegir el diseño e implementación de la aplicación basado en Android por la unión del lenguaje de programación utilizado en el trabajo actual como es Java y la amplia posibilidad de diseño gráfico que permite este sistema operativo.

En el desarrollo del proyecto se encuentran algunas dificultades a la hora de incluir nuevas funcionalidades a la aplicación inicial como por ejemplo la interconexión entre los dispositivos y la complicada depuración de código de la parte del dispositivo wearable.

Dejando de lado la parte de la programación pura y dura, se dedica mucho tiempo al diseño gráfico de la aplicación debido a la gran afición por la misma realizando todos los iconos y diseños de pantallas hasta el más mínimo detalle.

Como principal punto negativo del proyecto, destaca la excesiva extensión del mismo debido a la falta de tiempo por la gran carga de trabajo en el puesto actual y la dificultad que conlleva compatibilizar vida laboral y personal en el sector de la consultoría.

A pesar de las dificultades encontradas en la realización de este proyecto, la satisfacción de poder realizar una aplicación desde cero y obtener nuevos conocimientos sobre un sistema operativo tan utilizado es muy grande. De hecho, se desea en un futuro poder mejorar la aplicación y añadir nuevas funcionalidades como la utilización de una base de datos centralizada alojada en un servidor compartido donde puedan sincronizarse todos los integrantes del propio equipo.

Referencias y bibliografía

[1] Arquitectura Android:

PFC Universidad Carlos III de Madrid: Conceptualización, análisis, diseño e implementación de un videojuego estilo retro para plataforma Android. Autor: Manrique Martínez, Óscar. Julio 2013.

http://e-archivo.uc3m.es/handle/10016/18032

[2] Componentes de una aplicación Android (Fecha de consulta: Agosto 2015):

http://www.androidcurso.com/index.php/tutoriales-android/31-unidad-1-vision-general-y-entorno-de-desarrollo/149-componentes-de-una-aplicacion

[3] Patrón diseño MVC (Fecha de consulta: Septiembre 2015):

http://androideity.com/2012/05/10/la-importancia-del-mvc-en-android/

[4] Patrón diseño MVC:

PFC Universidad Politécnica de Cartagena: Aplicación para dispositivos móviles Android: Guía de los edificios de la Universidad Politécnica de Cartagena. Autor: Luis Gonzalo Soto Aboal. Agosto 2011.

http://repositorio.bib.upct.es/dspace/bitstream/10317/2670/1/pfc4240.pdf

[5] Android Studio (Fecha de consulta: Septiembre 2015):

http://developer.android.com/sdk/index.html

[6] Diseño vistas en Android Wear (Fecha de consulta: Octubre 2015):
http://developer.android.com/training/wearables/ui/lists.html

[7] Dispositivos inteligentes:

https://es.wikipedia.org/wiki/Tel%C3%A9fono_inteligente

https://es.wikipedia.org/wiki/Reloj_inteligente

[8] Jesús Tomás Gironés: El gran libro de Android (Editorial Marcombo, 3ª edición 2013).

[9] Android Developers (Fecha de consulta: Octubre 2015):

http://developer.android.com/index.html

[10] Android Wear (Fecha de consulta: Octubre 2015):

https://es.wikipedia.org/wiki/Android_Wear

[11] Bluetooth (Fecha de consulta: Octubre 2015):

https://es.wikipedia.org/wiki/Bluetooth

[12] Jesús Tomás Gironés, Vicente Carbonell: El gran libro de Android Avanzado (Editorial Marcombo, 1ª edición 2014)

[13] Wei-Meng Lee: Android 4 Desarrollo de aplicaciones(Editorial Anaya, 2012)

Glosario de términos

Smartphone: Teléfono móvil construido sobre una plataforma informática móvil, con una mayor capacidad de almacenar datos y realizar actividades, semejante a la de una minicomputadora, y con una mayor conectividad que un teléfono móvil convencional [7].

Smartwatch: Reloj de pulsera que ofrece funciones mejoradas ampliamente en relación a las de un reloj de pulsera habitual [7].

Android: Sistema operativo basado en Linux, diseñado principalmente para móviles con pantalla táctil como teléfonos inteligentes o tabletas inicialmente desarrollados por Android, Inc., que Google respaldó financieramente y más tarde compró.

Android Wear: Sistema operativo para dispositivos corporales (wearables) basado en Android que Google presentó a la sociedad el 18 de marzo de 2014. El sistema en sí está pensado para ser utilizado en relojes inteligentes (smartwatches), pulseras inteligentes (smartbands), y cualquier otro dispositivo wearable que pueda surgir en el futuro [10].

Aplicación: Forma de denominar a las aplicaciones informáticas para dispositivos móviles como son los Smartphone o las tabletas.

Aplicación Wear: Forma de denominar a las aplicaciones informáticas para dispositivos wearables como relojes y pulseras inteligentes.

Bluetooth: Especificación industrial para Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2,4 GHz [11].